DDC

AIR FORCE INSTITUTE OF TECHNOLOGY

LEVEL

AIR UNIVERSITY
UNITED STATES AIR FORCE

SCHOOL OF ENGINEERING

WRIGHT-PATTERSON AIR FORCE BASE, OHIO

AFIT/GE/EE/78D-31

(1)

LEVEL II

AD A064315

DDC FILE COPY

(6) MICROPROCESSOR-BASED DIGITAL AUTOPILOT DEVELOPMENT FOR THE XBQM-106 MINI-RPV.

(9) Master's THESIS

(14) AFIT/GE/EE/78D-31

(10) Olin D. Koger, Maj, USAF
Dale E. Tietz, Capt, USAF

(11) Dec 78 (12) 278p.

012 225

79 01 30 121 LB

AFIT/GE/EE/78D-31

MICROPROCESSOR-BASED DIGITAL AUTOPILOT DEVELOPMENT FOR THE XBQM-106 MINI-RPV

THESIS

Presented to the Faculty of the School of Engineering

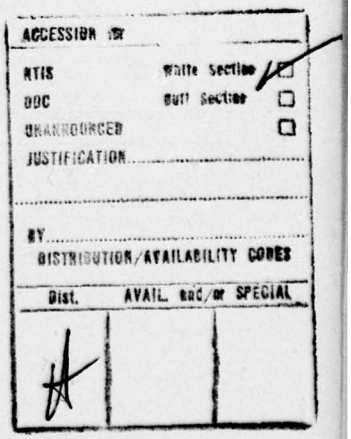
of the Air Force Institute of Technology

Air Training Command

in Partial Fulfillment of the

Requirements for the Degree of

Master of Science



by

Olin D. Koger, B.S.
Maj USAF

Dale E. Tietz, B.S.
Capt USAF

Graduate Electrical Engineering

December 1978

Approved for public release; distribution unlimited

Preface

This thesis set out to demonstrate that a microprocessor could be used as the central element in the autopilot for the XBQM-106 Mini-RPV. In addition to fulfilling this goal, we have shown that the autopilot can be constructed at ultra-low cost, a primary factor in the XBQM-106 program. Also, for the first time three important tasks have been accomplished: the data pertinent to this subject has been brought together in one place, a thorough literature search of the subject area has been made, and a complete control analysis of the factors relevant to the control of the XBQM-106 has been made. In short, a solid framework has been laid for follow-on development of a flight-capable autopilot and the incorporation of additional features into it.

We wish to thank the personnel of Flight Dynamics Laboratory's Mini-RPV Group for suggesting this project, providing the majority of the basic data needed, and giving freely of their time to assist in innumerable ways. We also would like to thank Dr. Gary B. Lamont for his overall guidance and encouragement. Finally, special thanks must go to Mr. Keith Jones of the ASD Computer Center Hybrid Simulation Branch. Without his help, verification of our results would have been impossible.

Olin D. Koger

Dale E. Tietz

Contents

List of Figures

List of Tables

List of Symbols and Definitions

General Definitions

Symbol	Mnemonic	
A_o	VSND	Speed of sound, ft/sec
a_y	AY	Lateral acceleration, ft/sec^2
b	SPAN	Wing span, ft
$\bar{C}$	CHORD	Mean aerodynamic chord, ft
C_1	---	Generalized autopilot rate gain
C_2	---	Generalized autopilot attitude gain
g	---	Gravity constant, 32.2 ft/sec^2
I_{xx}	IXX	Moment of inertia about roll axis (X), slugs-ft^2
I_{yy}	IYY	Moment of inertia about pitch axis (Y), slugs-ft^2
I_{zz}	IZZ	Moment of inertia about yaw axis (Z), slugs-ft^2
I_{xz}	IXZ	Product of inertia, slugs-ft^2
K_θ	KPA	Pitch attitude gain, dimensionless
K'_θ	KPAD	Pitch attitude gain, digital, dimensionless
$K^{\bullet}_\theta$	KPR	Pitch rate gain, sec
$K'^{\bullet}_\theta$	KPRD	Pitch rate gain, digital, dimensionless
K_ϕ	KRA	Roll attitude gain, dimensionless
K'_ϕ	KRAD	Roll attitude gain, digital, dimensionless
$K^{\bullet}_\phi$	KRR	Roll rate gain, sec
$K'^{\bullet}_\phi$	KRRD	Roll rate gain, digital, dimensionless
$K^{\bullet}_\psi$	KYR	Yaw rate gain, sec
$K'^{\bullet}_\psi$	KYRD	Yaw rate gain, digital, dimensionless
K_{ay}	KAY	Lateral acceleration gain (yaw loop)
K'_{ay}	KAYD	Lateral acceleration gain, digital, dimensionless
K_A	KA	Attitude gain, dimensionless

Symbol	Mnemonic	
K_{ACT}	KACT	Actuator gain sensitivity, deg/V
K_R	KR	Rate gain, dimensionless
K_{RS}	KRS	Rate sensor gain sensitivity, V/deg/sec
K_{VG}	KVG	Vertical gyro gain sensitivity, V/deg
K_{θ_c}	KTHC	Ground control pitch gain, dimensionless
K_{ϕ_c}	KPHC	Ground control roll gain, dimensionless
$K_{\dot{\psi}_c}$	KPSDC	Ground control yaw rate gain, dimensionless
M	MACH	Mach number
m	---	Mass, slugs
p	---	Roll rate, deg/sec
q	---	Pitch rate, deg/sec
$\bar{q}$	QDYN	Dynamic pressure, lb/ft^2
r	---	Yaw rate, deg/sec
s	---	Laplace operator
S	AREA	Wing area, ft^2
T_{max}	TMAX	Maximum thrust, lbs
---	THRST	Calculated cruise thrust, lbs
u	---	Forward perturbation velocity, ft/sec
U	---	Forward velocity, ft/sec
V, V_T	VFPS	Total velocity, ft/sec
Z_T	ZT	Thrust line offset, ft
α	AOA	Angle of attack, deg
β	---	Sideslip angle, deg
γ_o	GAMMA	Flight path angle, deg
ε_T	EPSTH	Thrust line inclination with respect to body X-axis, deg

Symbol	Mnemonic	
δ_a	DA	Aileron deflection, $\delta_a = \frac{\delta_{a_L} - \delta_{a_R}}{2}$, deg
δ_e	DE	Elevator deflection, deg
δ_r	DR	Rudder deflection, deg
ζ	ZETA	Damping ratio
θ	TH	Pitch attitude, deg
θ_c	THCMD	Pitch attitude command, deg
$\dot{\theta}$	THDOT	Pitch rate, deg/sec
ϕ	PHI	Roll attitude, deg
ϕ	PHCMD	Roll attitude command, deg
$\dot{\phi}$	PHIDOT	Roll rate, deg/sec
$\dot{\psi}_c$	PSDCMD	Yaw rate command, deg
ψ	PSI	Yaw attitude, deg
$\dot{\psi}$	PSIDOT	Yaw rate, deg/sec
ρ	DEN	Mass density of air, slugs/ft^3
ω_n	---	Undamped natural frequency, rad/sec
τ_r	---	Yaw rate washout time constant, sec

Aero Coefficients (Ref 31)

$C_{D_{MIN}}$	CDMIN	Minimum drag coefficient (C_L versus C_D curve)
$C_{L_{\alpha=0}}$	CL(AOA=0)	Lift coefficient at α=0
$C_{L_{C_{D_{MIN}}}}$	CL(CDMIN)	Lift coefficient at $C_{D_{MIN}}$ (C_L versus C_D curve)
$C_{L_{MAX}}$	CLMAX	Maximum lift coefficient (C_L versus C_D curve)
$C_{m_{\alpha=0}}$	CM(AOA=0)	Pitching moment coefficient at α=0
$K_{C_L}{}^2$	KCL2	Curve fit constant (C_L versus C_D curve)

Symbol	Mnemonic	
$K_{\delta_e}{}^2$	KDE2	Curve fit constant (drag polar), $1/deg^2$
$\delta_{e_{C_{D_{MIN}}}}$	DEMIN	Elevator deflection at $C_{D_{MIN}}$, deg

Stability Derivatives (Ref 29;30;31)

C_{D_M}	CDM	Change of drag with mach, 1/rad
C_{D_0}	CDO	Trim drag coefficient, 1/rad
C_{D_α}	CDA	Change of drag with angle of attack, 1/deg
$C_{D_{\dot{\alpha}}}$	CDAD	Change of drag with rate of angle of attack, 1/rad
$C_{D_{\delta_e}}$	CDDE	Change of drag with elevator deflection, 1/deg
C_{D_q}	CDQ	Change of drag with pitch rate, 1/rad
C_{L_q}	CLQ	Change of lift with pitch rate, 1/rad
C_{L_0}	CLO	Trim lift coefficient, 1/rad
C_{L_M}	CLM	Change of lift with mach, 1/rad
C_{L_α}	CLA	Change of lift with angle of attack, 1/deg
$C_{L_{\dot{\alpha}}}$	CLAD	Change of lift with rate of angle of attack, 1/rad
$C_{L_{\delta_e}}$	CLDE	Change of lift with elevator deflection, 1/deg
C_{l_p}	CLP	Change of rolling moment with roll rate, 1/rad
C_{l_r}	CLR	Change of rolling moment with yaw rate, 1/rad
C_{l_β}	CLB	Change of rolling moment with sideslip angle, 1/deg

Symbol	Mnemonic	
$C_{l_{\dot{\beta}}}$	CLBD	Change of rolling moment with sideslip angular rate, 1/rad
$C_{l_{\delta_a}}$	CLDA	Change of rolling moment with aileron deflection, 1/deg
$C_{l_{\delta_r}}$	CLDR	Change of rolling moment with rudder deflection, 1/deg
C_{m_q}	CMQ	Change of pitching moment with pitch rate, 1/rad
C_{m_M}	CMM	Change of pitching moment with mach, 1/rad
C_{m_α}	CMA	Change of pitching moment with angle of attack, 1/deg
$C_{m_{\dot{\alpha}}}$	CMAD	Change of pitching moment with angular rate of attack, 1/rad
$C_{m_{\delta_e}}$	CMDE(AOA)	Change of pitching moment with elevator deflection, 1/deg
C_{n_p}	CNP	Change of yawing moment with roll rate, 1/rad
C_{n_r}	CNR	Change of yawing moment with yaw rate, 1/rad
C_{n_β}	CNB	Change of yawing moment with sideslip angle, 1/deg
$C_{n_{\dot{\beta}}}$	CNBD	Change of yawing moment with sideslip angular rate, 1/rad
$C_{n_{\delta_a}}$	CNDA	Change of yawing moment with aileron deflection, 1/deg
$C_{n_{\delta_r}}$	CNDR	Change of yawing moment with rudder deflection, 1/deg
C_{y_p}	CYP	Change of sideforce with roll rate, 1/rad
C_{y_r}	CYR	Change of sideforce with yaw rate, 1/rad

Symbol	Mnemonic	
C_{y_β}	CYB	Change of sideforce with sideslip angle, 1/deg
$C_{y_{\dot{\beta}}}$	CYBD	Change of sideforce with sideslip angular rate, 1/rad
$C_{y_{\delta_a}}$	CYDA	Change of sideforce with aileron deflection, 1/deg
$C_{y_{\delta_r}}$	CYDR	Change of sideforce with rudder deflection, 1/deg

Abstract

The development of a digital autopilot for the XBQM-106 Mini-Remotely Piloted Vehicle (RPV) is described. The design is based on the RCA CDP-1802 microprocessor with ultra-low component cost as a primary goal. The digital autopilot is developed through the breadboard stage, and performance testing is accomplished via hybrid simulation. First, aircraft transfer functions are derived from aerodynamic wind tunnel data. Then a complete control analysis is made of the existing analog autopilot, sensors, actuators, and aircraft dynamics. This is followed by synthesis of the digital algorithms (including a digital filter) by means of digitization. Hardware is selected based on system requirements, and hardware design details are included. Software to execute the digital algorithms is developed and described. The results of testing by hybrid simulation show that the performance of the digital autopilot is comparable to the existing analog autopilot. Finally, recommendations for the follow-on study are included.

MICROPROCESSOR-BASED DIGITAL AUTOPILOT DEVELOPMENT FOR THE XBQM-106 MINI-RPV

I. Introduction

The past few years have brought about rapid advances in the field of microprocessor technology and applications. Impressive economies associated with this technology can be achieved and include such benefits as increased performance, flexibility, reliability, maintainability, modularity, density, and decreased component costs. For control applications, digital processing circuits demonstrate far less drift with respect to such factors as temperature, voltage, age, etc., than analog circuits, and result in very precise repeatability from unit to unit. The microprocessor and its constituent analog/digital circuitry is now the vogue in design engineering (Ref 6;45).

Statement of Problem

The Mini-RPV (Remotely-Piloted Vehicle) Group of the Air Force Flight Dynamics Laboratory (AFFDL) at Wright-Patterson AFB suggested that an investigation be conducted into the feasibility of designing a prototype digital autopilot (A/P) for the XBQM-106 Mini-RPV. The A/P should be microprocessor-based, emulate the already existing KBG Corporation analog A/P, and be designed with ultra-low component cost in mind. Performance of the digital A/P should be compared to that of the analog A/P which has been modelled on the Comcor/Sigma 7 Hybrid Simulator at the WPAFB Computer Facility. Actual flight testing of the re-

sultant A/P should not be considered but could be included in a follow-on study.

After evaluating the nature of the problem, it was determined that the analog A/P and its associated control loop characteristics had been developed by indirect "hand-tweaking" of gains based on simulation and flight testing. No conventional control loop analysis had been accomplished to validate the aircraft and A/P performance. In addition, concrete performance characteristics had not been established.

Because of this added aspect, a complete conventional control analysis of the various loops is needed to verify the already established loop gains and to provide a vehicle for further performance evaluations. Providing this analysis will greatly enhance the understanding of the RPV and A/P dynamics.

Approach

Evaluation of the recommended proposal, stated in a letter from the Mini-RPV Group, yields a basic framework for the development of this study.

Chapter II develops the necessary linear system model transfer functions for the RPV, sensors and actuators. These dynamic characteristics are evaluated from a frequency standpoint for possible inclusion into linear control loop models representative of the 3-axis analog A/P. Chapter II references Appendix A which contains an extensive aircraft transfer function derivation section. For documentation purposes, time responses, frequency responses, and root loci are included for these functions. In addition, a summary of simulated free RPV time re-

sponses are attached for comparative purposes. The results of this chapter provide the additional framework for control analysis in Chapter III.

Chapter III describes the analog A/P and its functional characteristics. Based upon the constraints imposed by this system, a 3-axis control loop model is generated for the analysis. Transfer functions are incorporated from Chapter II. Since the primary purpose of this chapter is to develop a reliable linear system model of the A/P suitable for control analysis, the measure of accuracy or validity is determined by comparing the time responses of the model A/P, axis to axis, with the simulated analog A/P. Similar forcing functions are assumed. In order to manipulate the model to produce these similar qualitative results, the actual feedback gains employed in the analog A/P are used directly in the linear model. No attempt is made to optimize loop performance by "gain-tweaking" or compensation. Root locus and time response techniques are used in the analysis. A spin-off of this analysis yields recommendations for improved A/P performance.

In Chapter IV the control loops of the analog A/P are digitized and digital feedback controllers established. The important factors associated with digital control are investigated. These include: word length, sampling frequency, transport delay and numerical errors. The analog high-pass filter in the yaw rate loop is converted, via the Tustin transformation, into a digital filter represented as a linear difference equation. Once the digital A/P model is developed, the next two chapters proceed with the digital A/P hardware and software design.

Consistent with the already established digitization scheme, Chapter V describes the actual digital A/P hardware design to include microprocessor and component selection, system build-up, functional breadboarding of the circuitry and troubleshooting. Schematics illustrate in detail the progressive A/P mechanization.

Chapter VI develops the required digital control software and operating system for A/P operation. One axis at a time is considered, tested, then incorporated into the main software program. The final product consists of an interleaved, interrupt-oriented 3-axis control algorithm with digital filtering requiring approximately 650 eight-bit bytes of memory. Flow charting, testing, and evaluations are included for support.

To graphically demonstrate the comparative performance characteristics of the analog and digital A/P's, Chapter VII analyzes the results of time response testing via hybrid simulation. Qualitative examination reveals very close agreement both in the transient and steady state. Sources of errors are discussed. In addition, time response histories versus sampling frequency for the digital A/P show progressive performance degradation for decreasing frequencies. A feasible range of sampling frequencies is thus established.

Chapter VIII discusses potential areas of improvement in the analog A/P. Recommendations include stating specific A/P performance criteria, re-evaluating the dynamic characteristics of the RPV actuators because of their overly restrictive nature, and investigating the inclusion of feedback compensation in the control loops to enhance performance. Recommendations for the digital A/P include a follow-on study incorporating

additional A/P modes (altitude hold, heading hold), actual flight testing, and finally investigating the feasibility of using the new RCA 1804 microprocessor in a more compact flight-testable version of the current system. Final remarks conclude this study and provide the motivation for its future continuance.

Literature Search

A thorough literature search indicates this investigation to be the only current DOD study of its type germane to Mini-RPV's. Only one other study, sponsored by Melpar Corporation, centers on this subject, but their results, based on the Fairchild F-8 microprocessor, are sketchy and proprietary (Ref 33). A list of digital A/P studies particularly relevant to this investigation include:

--Northrop Ventura (MQM-74C) digital drone based upon the sixteen-bit TI 9900 microprocessor; proprietary; most useful; dated 15 Jan 1978 (Ref 6;45)

--Numerous missile digital A/P's from 1973 to present; technology old; control analysis still valid (Ref 2;20;26)

--Teledyne Ryan (M-262) Navy Mini-RPV with combination analog/digital control system; informative, non-technical; technology old; dated 24 Oct 1975 (Ref 46)

These few references contain the most up-to-date information available on specific applications of microprocessor-based digital flight control/A/P's to drones, missiles and Mini-RPV's. Since the interest generated in this field appears to be mushrooming, this study provides a "bench mark" for further Mini-RPV digital A/P developments.

II. System Model

In order to easily understand and evaluate a particular system, it must be modelled in a manner which is both mathematically tractable and accurate. This model representation must approximate closely the "real world" non-linear system. With this in mind, this chapter proceeds to model the RPV, sensors, actuators, and 3-axis A/P in a manner suitable for conventional linear control analysis. The definition of A/P includes all sensors, actuators, and control loops needed to accomplish desired aircraft stabilization. RPV system modelling includes the following:

---general aircraft description

---general A/P model

---RPV transfer function derivation

---sensor and actuator characteristics

---complete system model

General Description

The XBQM-106 aircraft, as depicted in fig. 2-1, is a lightweight propeller-driven vehicle---termed a Mini-RPV (miniature remotely-piloted vehicle). This specific nomenclature is adopted from weight and size parameters described in the AFFDL document RPV Flying Qualities Design Criteria (Ref 41). Primary usage of this vehicle is:

---a technology demonstrator (test bed for testing mission profiles and equipment)

---enemy harassment in hostile environment (flown either under drone program control or as a manually controlled data-linked RPV)

Figure 2-1: XBQM-106 Mini-RPV

General aircraft characteristics include:

--maximum weight - 200 pounds loaded

--wing span - 10 feet

--fuselage length - 10 feet

--engine - 10 HP McCulloch

--service ceiling - 10,000 feet (approx.)

Additional aircraft parameters may be found in (Ref 29;39).

For RPV flight orientation a fixed axis system convention is used throughout and is described by Euler angles (eg. θ, ϕ, ψ) expressed in degrees (see fig. 2-2) (Ref 32:208). Since control surface deflection terminology (eg. δ_e, δ_a, δ_r) also plays an important role in the understanding of time responses, fig. 2-3 is included as a visual aid.

Conventions are as follows for positive sense:

δ_e : Elevator deflection; trailing edge down (TED) → positive

δ_a : Aileron deflection; $\delta_a = (\delta_{a_L} - \delta_{a_R})/2$

where

δ_{a_L}: Lefthand aileron deflection; TED → positive

δ_{a_R}: Righthand aileron deflection; TED → positive

δ_r : Rudder deflection; trailing edge left → positive

Autopilot Description

Two types of flying analog A/P's are in existence, one manufactured by General Dynamics (Pomona) and the other developed by KBG Corporation (in-house avionics contractor, Mini-RPV Group, AFFDL, WPAFB). This study will concern itself only with the latter A/P since the General Dynamics unit was designed for use with a specific target seeker (Ref 29).

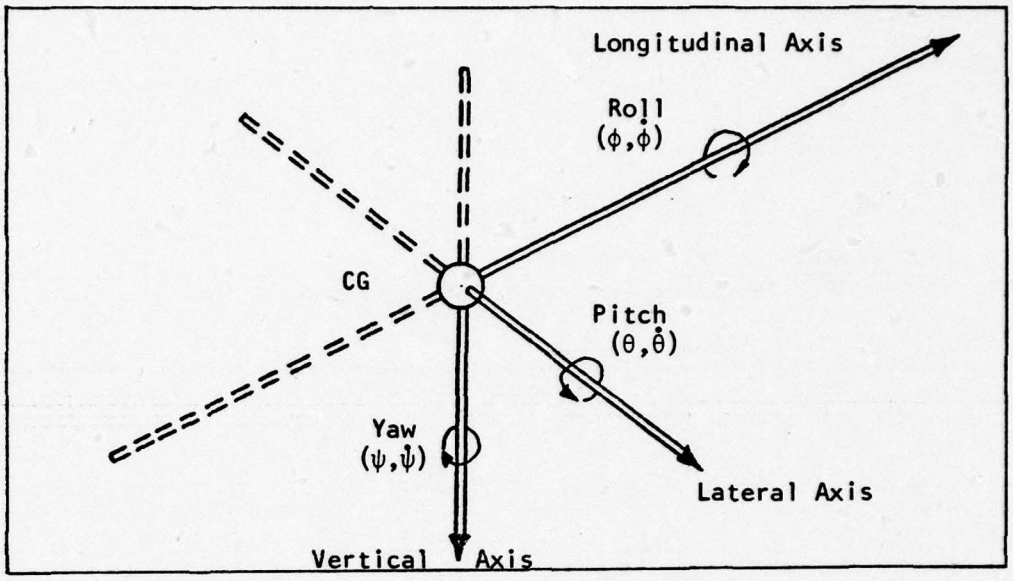


Figure 2-2: Fixed Axis Conventions (Positive Sense)

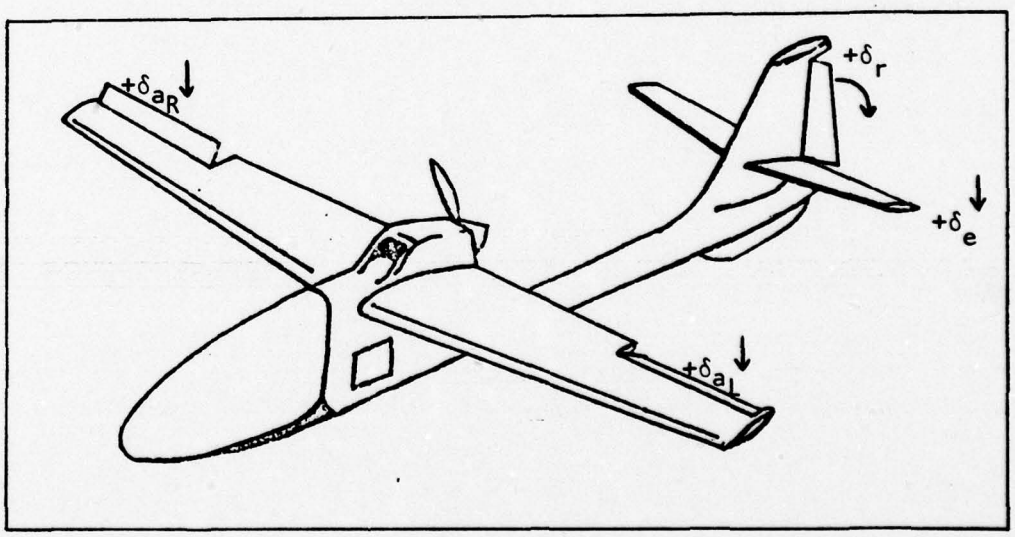


Figure 2-3: Control Surface Deflection Conventions (Degrees)

The detailed schematics for the analog A/P are illustrated in (Ref 29) and need not be presented here. More relevant, however, is the generalized A/P block diagram in fig. 2-4.

At present, the analog A/P is flown in two distinct modes: manual ground control and free-flight stabilizer.

As shown in fig. 2-4, manual pulse-width modulated (PWM) ground control commands direct the aircraft to a desired flight attitude. With the A/P engaged, this "fly-through" capability allows for improved stability. Data-linked commands are mixed with analog A/P outputs in a pulse-width/analog mixer (essentially a summing junction), and then directed, in PWM form, to the various control surface actuators.

In the absence of any ground commands, the A/P functions as a wing-levelling 3-axis stabilizer. Disturbances such as gusts or down-drafts are sensed by the A/P, which in turn reacts to return the aircraft to its original attitude. This mode is very important for free-flight cruise operation.

Note that those items starred (*) in the diagram are future operational modes and are not presently incorporated into the A/P.

A generalized 3-axis analog A/P model is depicted in fig. 2-5, and represents the actual analog A/P control system. This design is conventional and noted in several references (Ref 4; 17; 32). The model takes into consideration the following set of important system constraints:

--All loop gains are expressed as constants in the <u>feedback paths</u> only. There are no <u>feed-forward</u> gains.

--Rate and attitude loop controllers are placed in the <u>feedback paths</u> only.

--The yaw damper includes rate and lateral acceleration feedback.

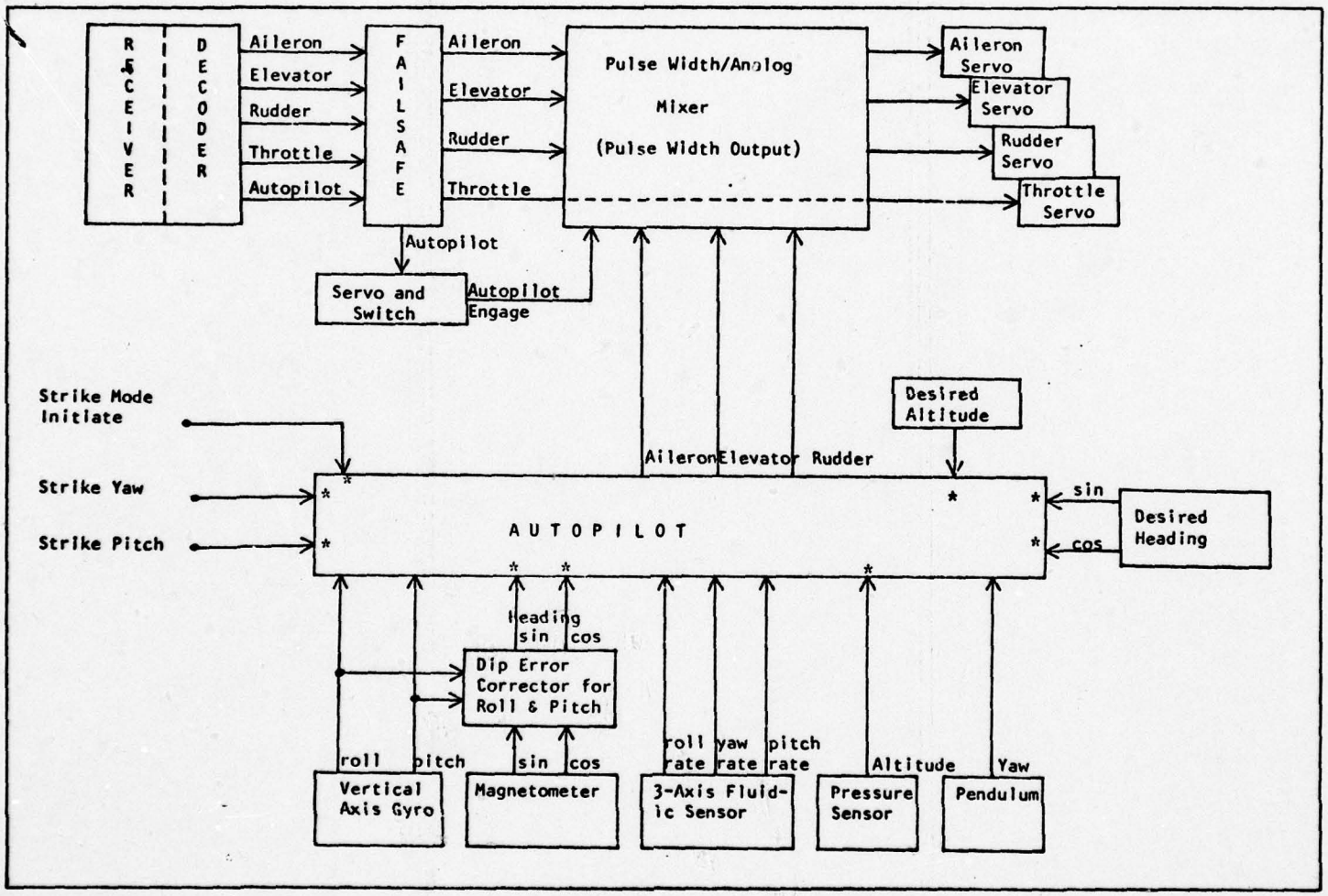


Figure 2-4: General Analog Autopilot

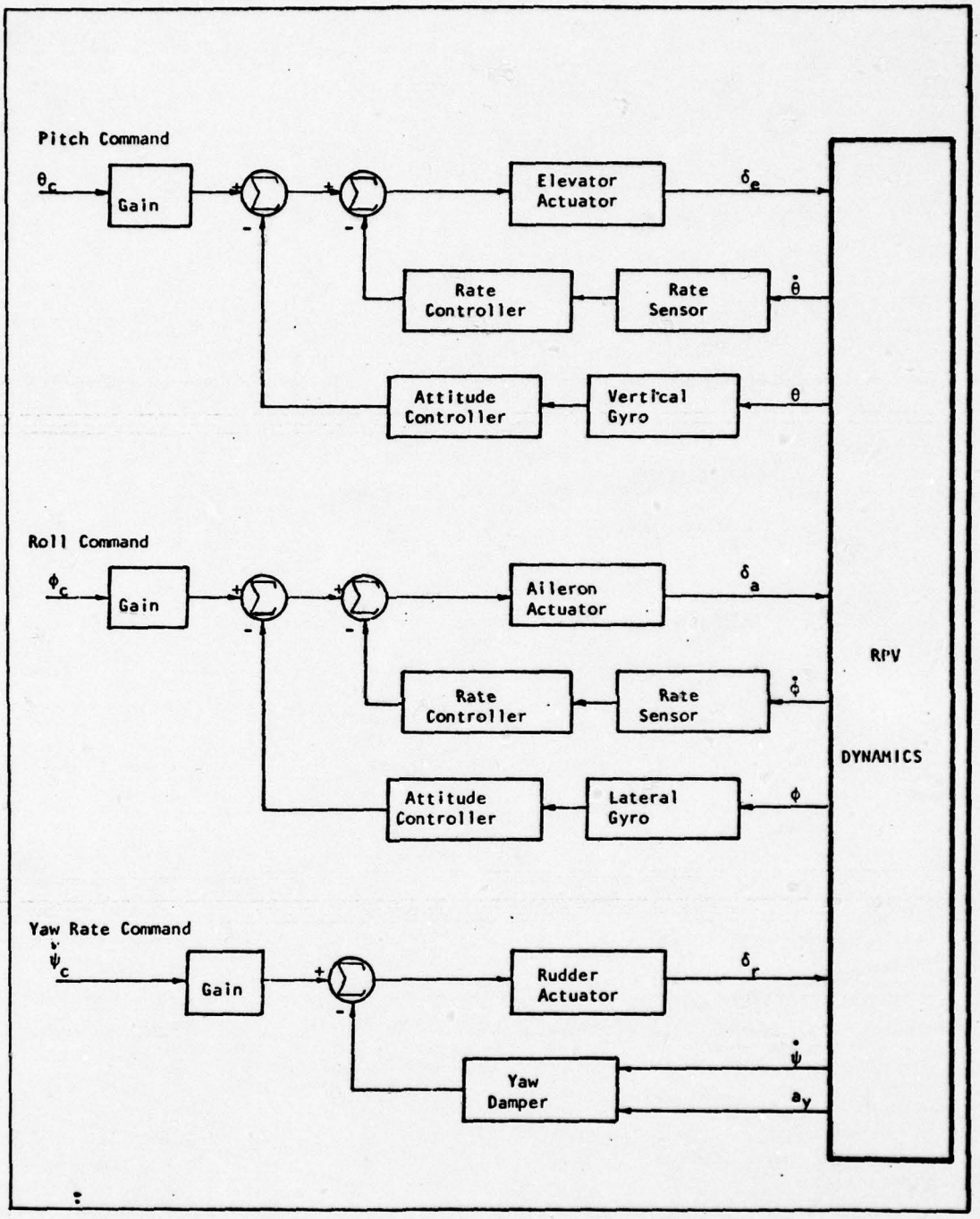


Figure 2-5: General Analog Autopilot Model

--The flying A/P has a pulse-width/analog mixer depicted in fig. 2-4. The PWM scheme can be approximated as a continuous system (i.e. unity gain) and neglected from further consideration (Ref 20).

As shown in fig. 2-5, the A/P stabilizes the three independent axes of pitch, roll and yaw. Both pitch and roll have rate and attitude feedback whereas yaw has rate and lateral acceleration feedback. Sensors on the RPV detect such parameters as angular rate, attitude, and acceleration, and feed these signals into respective loop controllers. The controller outputs are then summed with data-linked commands, if any, and then routed to the control surface actuators. Thus, closed-loop negative feedback stabilization is accomplished (Ref 29).

Now that a conceptual model is described, it is necessary to determine the respective Laplace transfer functions for the RPV, sensors and actuators. These are required "building-blocks" for further analysis.

Aircraft Transfer Functions

The first major requirement to model a control system similar to fig. 2-5 is to ascertain the necessary aircraft transfer functions. Since these were not available, they had to be derived. Using current aerodynamic wind tunnel data (Ref 39) and an interactive computer program (Ref 30;31), called the McGlynn program, linearized transfer functions for all three axes were determined for three specific flight regimes (see Appendix A for detailed derivation and listings).

	Altitude (ft, msl*)	Weight (lbs)	Mach
Case I	1000	135	.125
Case II	3000	135	.125
Case III	6000	135	.125

*Mean Sea Level

These three cases were chosen to investigate the sensitivity of the transfer function coefficients to changes in altitude while keeping weight and mach constant. A visual inspection of these functions reveals an insignificant change in the coefficient values. Such small deviations indicate relative stability of the functions with respect to altitude changes. As such, the prospect of having to employ "gain scheduling" when designing the A/P for all RPV operating altitudes is negated. Based upon this brief analysis, the transfer functions associated with Case II are chosen for use throughout the rest of the study. Time-responses, frequency responses, and root loci for all Case II functions are included in Appendix A. In addition, Appendix A contains hybrid simulation free-airframe RPV time responses to forcing functions nearly identical to those used in producing the transfer function time responses. In doing so, a direct means of comparison is made of the resultant waveforms. Very close agreement exists between transient and steady-state responses. This qualitative analysis provides a good measure of confidence to the modelling process, but is not an overriding issue at this point in the analysis.

Since aircraft dynamics are non-linear in nature, linear transfer function theory makes the following assumptions (Ref 4:21;30:5):

--aircraft is rigid

--earth is the inertial reference frame

--atmosphere is stationary or translating at a constant velocity relative to the earth

--vehicle mass, inertia, and angular momentum due to spinning rotors are constant

--vehicular motion consists of small disturbances (perturbations) about a reference steady-state flight condition (equilibrium)

--airflow is quasi-steady

The resulting linear equations are more tractable than their non-linear counterparts, and do not unduly restrict the control analysis. Strict adherence to control surface deflection and angle notation is maintained to provide continuity. Once the RPV transfer functions are established, sensors and actuators must be modelled.

Sensor and Actuator Characteristics

In order to model the complete control system, the individual component sensor and actuator dynamics must be known or estimated. As mentioned earlier, the current A/P consists of a vertical gyro (pitch/roll attitude), fluidic rate sensors (pitch/roll/yaw rates), an optional lateral accelerometer (yaw turn coordination), and actuators. Because these devices are low-cost and not of MIL-SPEC quality, little is known or published about actual dynamics. As a result, it was necessary to derive or approximate each component transfer function. The rate sensors and actuators required laboratory testing to determine their frequency characteristics. The lateral accelerometer (pendulum) transfer function was stated erroneously by the laboratory, but subsequent review corrected the mistake. The vertical gyro dynamics were specified in the literature (Ref 51). The following discussion provides an in-depth analysis and documentation of device dynamics and pertinent characteristics from which individual component models are extracted for use in the overall control model.

Vertical Gyro. The most critical and expensive ($2400) sensor is the commercially available Humphrey vertical gyro (VG34-0201-1) (Ref 51). It provides accurate pitch and roll attitude orientation for the aircraft. Compared to the aircraft natural frequencies, the gyro dynamics are sufficiently high to be modelled as a pure gain of unity. Justification for this assumption is based upon the fact that the erection system natural frequency (ω_{ne}) must be much less than the aircraft longitudinal phugoid natural frequency (ω_{np}) (Ref 42:10.25). Specification data indicates a nominal erection rate of 9 deg/min which in turn equates to ω_{ne} = .0026 rad/sec.

(9 deg/min)(1 min/60 sec)(1 rad/57.3 deg) = .0026 rad/sec

Since ω_{np} = .33 rad/sec for the current flight condition, the above assumption is valid.

Vertical gyro characteristics include (Ref 51):

---roll displacement (ϕ)	± 90 deg (± 1 deg)
---pitch displacement (θ)	± 60 deg (± 1 deg)
---linearity	± 1% full scale
---normal erection rate	9 deg/min (± 3 deg/min)
---drift	1 deg/min max. after 6 min of 3 deg reverse

Vertical gyro sensitivities include (Ref 29):

---pitch attitude	± 2.5 volts/± 60 deg = .04167 volts/deg
---roll attitude	± 2.5 volts/± 90 deg = .02778 volts/deg

Note: Because this device is so much more expensive than the other sensors, a comparable performance substitute is being sought by the laboratory.

Rate Sensors. A low-cost ($250) independent 3-axis fluidic rate sensor has been developed in the laboratory especially for RPV A/P applications. The device exhibits simple construction and excellent performance (Ref 29;39). Device dynamics were approximated by recording a rise time of 45 msec to a unit step input. The transfer function can be modelled as a first order lag with a cutoff frequency (ω_{n_r}) of 50 rad/sec as given by (Ref 29;39):

$$\frac{1}{\left(\frac{S}{\omega_{n_r}}\right) + 1} = \frac{50}{S + 50} \qquad (2\text{-}1)$$

The cutoff frequency is derived from the definition of rise time, which is the time required for the response to a unit step function input to rise from 10 to 90 percent of its final value (Ref 18:183). Assume a normalized first order lag response to a unit step, as a function of the time constant T and time t, to be expressed as (Ref 32:59):

$$1 - e^{-\left(\frac{1}{T}\right) t} \qquad (2\text{-}2)$$

By letting .045 seconds represent the 90 percent point, as illustrated in fig. 2-6, manipulation of Eq (2-2) yields a time constant of about 2.303T.

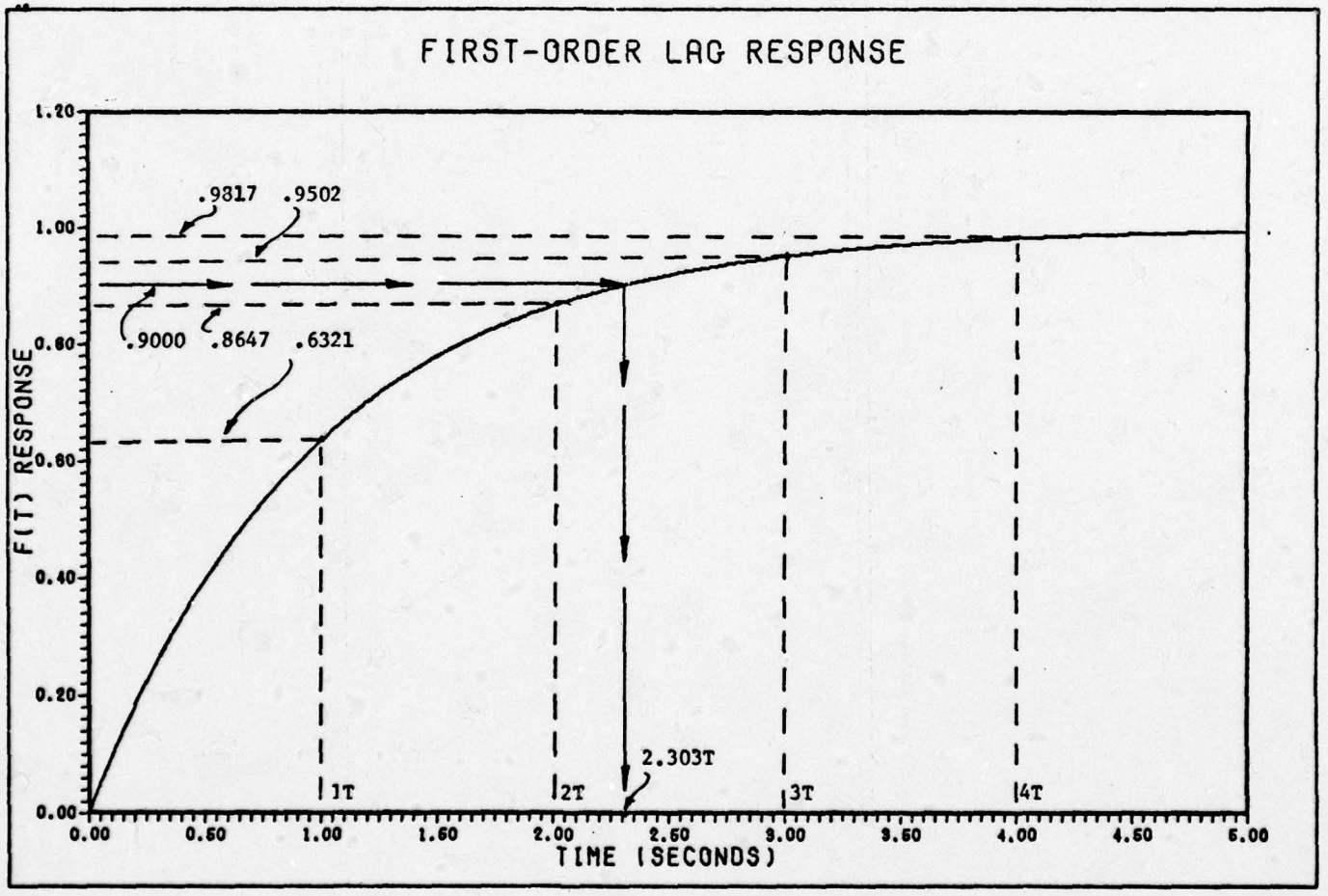


Figure 2-6: Normalized First-Order Lag Response-Rate Sensor

$$1 - e^{\frac{-t}{T}} = .90$$

$$e^{\frac{-t}{T}} = .1$$

$$t = 2.303T$$

Since .045 sec represents 2.303T, the time for one T is found by the expression:

$$\frac{.045 \text{ sec}}{2.303\ T} = \frac{X}{T}$$

Solving for X yields .0195 seconds. Since cutoff frequency (rad/sec) is the inverse of the time constant (sec), inverting .0195 sec yields 51.3 rad/sec (approximately 50 rad/sec).

Because the dynamics of the rate sensor exceed that of the aircraft longitudinal short-period by a factor of nearly seven (50 rad/sec vs. 7 rad/sec), the sensor can be neglected in the control analysis and incorporated in the rate loop as a unity gain.

Rate sensor characteristics include (Ref 29;39):

--good sensitivity range (maximum)	0 to ± 100 deg/sec
--noise	0.002 volts RMS
--excellent linearity	± 0.25%
--small size	88 mm X 88 mm X 63 mm
--low power consumption	3 watts at 28 VDC
--power required	13-28 VDC
--3 independent axes (pitch, roll, yaw) in one package	
--no hysterisis	

--no dead band

--good frequency response

Rate sensor sensitivities include (Ref 29;39):

All 3 axes: (± 2.5 volts/± 50 deg/sec) = .05 volts/deg/sec

Pendulum Accelerometer. For coordinated turning performance a commercially available, low-cost ($100) pendulum accelerometer is employed in the yaw axis. A feedback signal proportional to lateral acceleration in the y-direction is produced. Because the aircraft will eventually be flown in an uncoordinated yaw-to-turn mode, with wings level, this device will most likely be excluded from the A/P. For documentation purposes only its characteristics are included.

Specification sheet data reveals the Humphrey (CP17-0601-3) pendulum to be highly accurate and long-lived with a natural frequency ($\omega_{n_{p_a}}$) of 20.1 rad/sec (3.2 HZ) (Ref 38). Assuming first order lag dynamics, the transfer function is modelled as:

$$\frac{1}{\left(\frac{S}{\omega_{n_{p_a}}}\right) + 1} = \frac{20}{S + 20} \qquad (2\text{-}3)$$

Note: this transfer function had been erroneously entered as $\frac{1}{S + 2}$ in (Ref 29).

Characteristics include:

--accuracy ± 1% full scale

--resolution less than .2°

--linearity within .5%

--output range $\pm 45^{o}$

Note that the output in degrees is proportional, not to sideslip angle, but to angular acceleration (a_y) in ft/sec^2.

Device sensitivity is:

$$\frac{\pm 2.5 \text{ V}}{\pm 45^{o}} = .0556 \text{ volts/deg}$$

but

$$\pm 45^{o} \propto \pm 1 \text{ g} = \pm 32.2 \text{ ft/sec}^2$$

so in an alternative form,

$$\frac{\pm 2.5 \text{ V}}{\pm 32.2 \text{ ft/sec}^2} = .0776 \frac{\text{volts}}{\text{ft/sec}^2}$$

Actuator. Several commercial actuators were investigated for possible use in the aircraft. Based upon ruggedness, high speed, high torque characteristics, and especially low cost ($39), the Futaba FP-S14 actuator was finally chosen. Identical actuators are employed for deflection of primary flight control surfaces: elevator, aileron, rudder (Ref 29;39).

Each actuator is controlled by pulse-width modulated signals in the range of 650 μsec ~ 1900 μsec, 5 VDC nominal amplitude, with maximum torque of 13 kg-cm. Internal characteristics dictate a slew rate limitation on rotary angle of $\pm 45^{o}$ in .5 sec. at 5 VDC. This implies a maximum stop-to-stop actuator speed of approximately 90-100 deg/sec of

unrestricted deflection (Ref 44).

Since manufacturer device dynamic data was non-existent, laboratory frequency response testing was initiated on the actuator using an available phase-angle voltmeter connected to an x-y sweep plotter. Various plots were recorded as a function of load. The most representative case is depicted in fig. 2-7 with a load of 5.5 in-lb and a maximum servo output arm deflection of $\pm 20^{o}$. Overall results indicate that the actuator is essentially linear up to and including ± 20 degrees of deflection at varying loads. Non-linear behavior is exhibited for larger deflections. Control analysis must be restricted to this linear range.

By assuming second-order characteristics of fig. 2-7, a cutoff frequency (ω_{n_a}) is estimated to be 2.5 HZ (15.7 rad/sec) at the -90^{o} phase point. Subsequent magnitude and phase plot curve fitting through the use of TOTAL (Ref 28) yields the following actuator dynamics:

$$\frac{\delta(S)}{\delta_c(S)} = \frac{1}{\left(\frac{S}{\omega_{n_a}}\right)^2 + \frac{2(\zeta)S}{\omega_{n_a}} + 1} = \frac{246.5}{S^2 + 25.12S + 246.5} \qquad (2\text{-}4)$$

where

$\delta(S)$ = control surface deflection (deg)

$\delta_c(S)$ = control surface command (deg)

ω_{n_a} = natural cutoff frequency (rad/sec)

ζ = damping ratio

Fig. 2-8 exhibits the linear frequency response of the actuator from Eq (2-4). Actuator dynamics are sufficiently close to those of the basic aircraft to warrant inclusion in the control analysis model. Note, how-

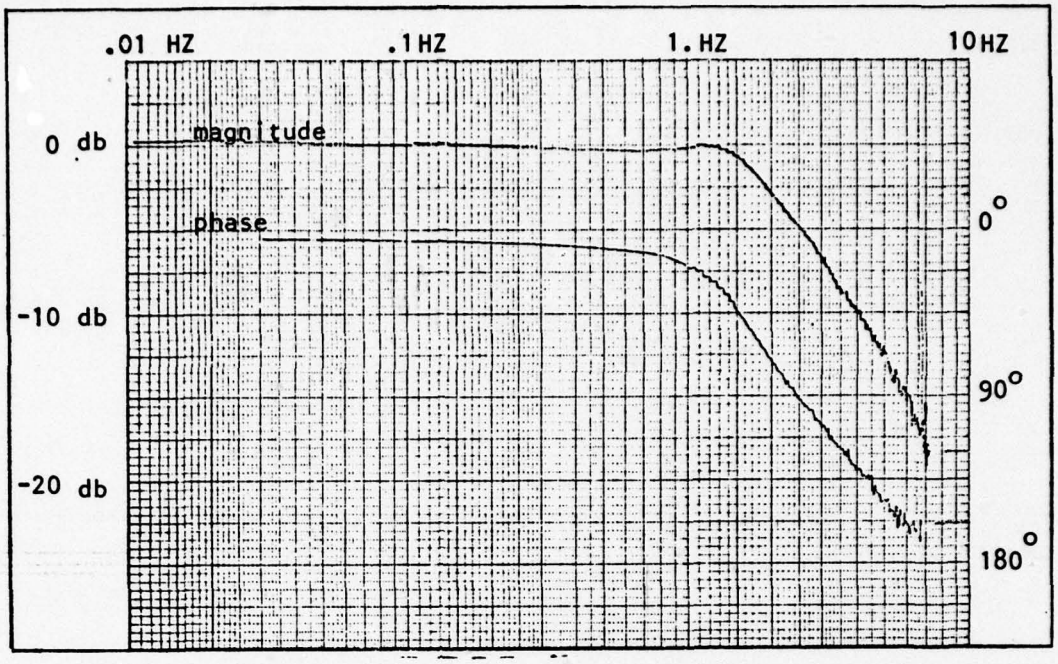


Figure 2-7: Experimental Actuator Frequency Response

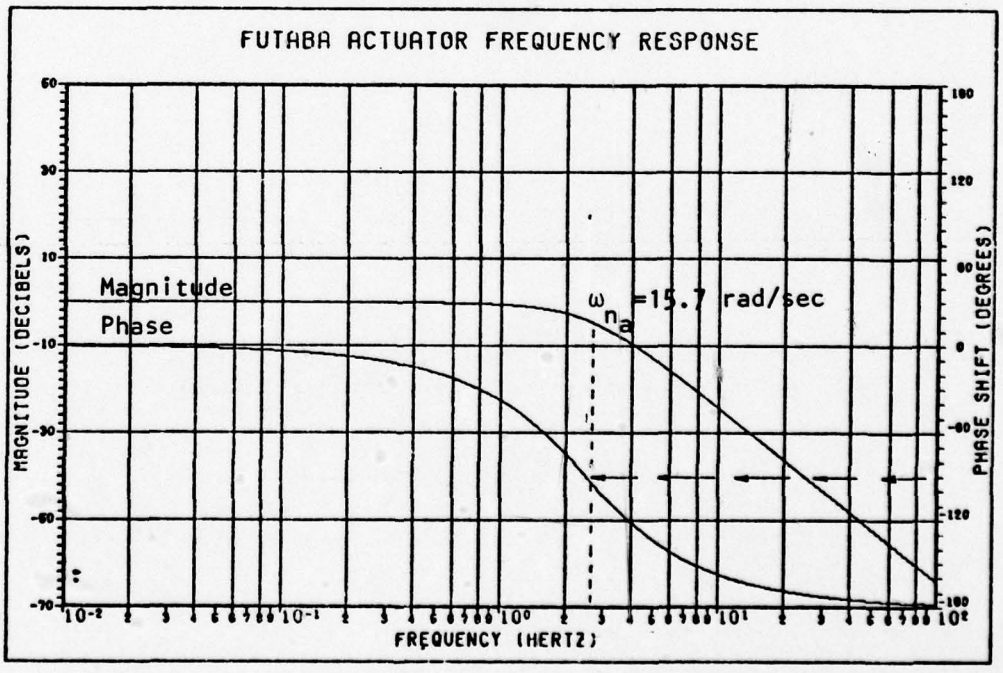


Figure 2-8: Model of Actuator Frequency Response

ever, that the actuator poles are complex conjugates; this will later be shown to impose constraints on the root-locus stability analysis.

Actuator rate limitation: ± 45 deg/sec

Control surface deflection angle limitations (mechanical):

δ_e (elevator)	± 8 deg
δ_a (aileron)	+ 14 deg (TEU)*
	- 11 deg (TED)*
δ_r (rudder)	± 16 deg

* TEU (trailing edge up)
TED (trailing down)

Complete System Model

Once the transfer functions for the aircraft, sensors, and actuators have been modelled and determined to be relevant to the control analysis, the analog A/P loop analysis can proceed. Chapter III models and analyzes the pitch, roll, and yaw axes independently and compares results with hybrid simulations. The qualitative analysis of the model with the simulation lends credence to modelling accuracy.

III. System Analysis

Having established the transfer functions for the components of the general A/P model (fig. 2-5, Chapter II), linear control analysis of each individual axis proceeds. The primary means of determining whether or not the model of the A/P performs like the analog A/P is through a qualitative comparison of transient responses generated by each axis of the A/P to the same input forcing function. Time responses for the model A/P are obtained from the computer program TOTAL (Ref 28) whereas those from the analog A/P result from the already existing hybrid simulation.

The key point of this comparison process lies in the need to directly substitute the analog A/P loop gains, obtained by indirect "hand-tweaking" methods (Ref 29) into the feedback loop gains of the model A/P. In order to do this, the analog gains require manipulation so as to be compatible with the model gain structure. Appendix B addresses the gain manipulation. It is important to stress that the model A/P performance is not to be optimized in any sense because this will negate the whole purpose of this comparative approach. Extensive use of the computer-aided-design program TOTAL (Ref 28) is employed throughout the analysis for generation of root loci, frequency responses and time responses.

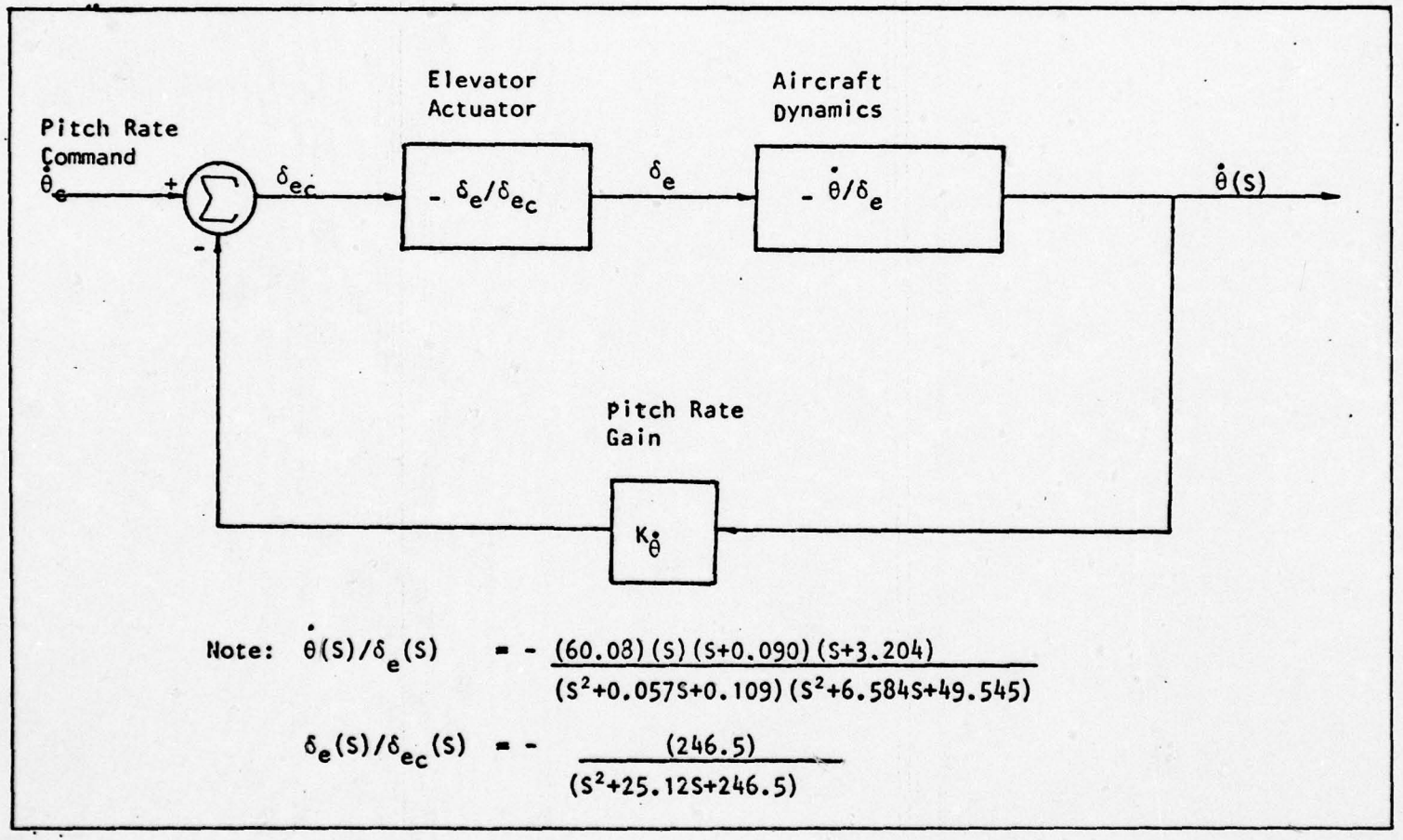
Longitudinal Controller

Based upon previous aerodynamic and simulation studies conducted by the Mini-RPV Group, the need to stabilize the longitudinal axis of the aircraft, when not operating under remote control, has been con-

firmed (Ref 29). Two dominant dynamic characteristics govern longitudinal stability for most subsonic aircraft: the "phugoid", and "short-period" oscillations (Ref 4:35). The "phugoid" or long-period" mode is generally more oscillatory (relatively undamped) in nature and of low natural frequency (ω_p). It consists primarily of variations in pitch attitude (θ), and forward velocity (U) with angle of attack (α) nearly constant. The "short-period" mode, on the other hand, is considerably more damped and of higher natural frequency (ω_{sp}). Variations occur in α and θ, but with very little change in U (Ref 4:35). If the reader wishes to vary these modes, refer to Appendix A, fig. A-1a ~ A-3b. A pilot flying an aircraft can usually control the "phugoid", but the "short-period" is much faster, and is often responsible for "pilot-induced-oscillations" (PIO) (Ref 42). Since the RPV must have the capability to fly without remote pilot control, both modes are included in the transfer functions for analysis.

This section first develops and analyzes the pitch rate controller, then develops the pitch attitude controller.

Linear Pitch Rate Controller. Control of pitch rate is provided by the linear rate controller shown in fig. 3-1. Pitch rate is detected by the rate sensor which is modelled as a unity gain transfer function (see chapter II). The signal is then multiplied by a suitable feedback gain ($K_{\dot{\theta}}$, KPR), fed back to a summer and subtracted from the pitch rate command ($\dot{\theta}_c$). The resulting error signal (δ_{e_c}) then drives the elevator actuator to the desired deflection angle (δ_e). Thus, pitch rate damping (control) is accomplished. In the absence of a pitch rate command, rate



Note: $\dot{\theta}(S)/\delta_e(S) = -\dfrac{(60.08)(S)(S+0.090)(S+3.204)}{(S^2+0.057S+0.109)(S^2+6.584S+49.545)}$

$\delta_e(S)/\delta_{e_c}(S) = -\dfrac{(246.5)}{(S^2+25.12S+246.5)}$

Figure 3-1: Block Diagram of Linear Pitch Rate Loop Controller

perturbations (e.g., gusts) applied externally to the aircraft will still be controlled via the feedback loop.

This study assumes that coupling of roll and yaw into the pitch axis and vice versa is negligible (Ref 4).

The analysis begins by multiplying the actuator and RPV transfer functions together to obtain the open-loop transfer function (OLTF):

$$\frac{\dot{\theta}\ (S)}{\delta_{e_c}\ (S)} = \frac{(14809.47)\,(S)\,(S+0.090)\,(S+3.204)}{(S^2+0.057S+0.109)\,(S^2+6.584S+49.545)\,(S^2+25.125S+246.5)} \qquad (3\text{-}1)$$

Since the sign of the RPV transfer function is negative, the sign of the actuator transfer function is also made negative to make the overall sign of Eq (3-1) positive. This is a standard method in pitch axis analysis (Ref 4:58). The resultant root loci for Eq (3-1) are shown in figs. 3-2a,b. Fig. 3-2a illustrates actuator and "short-period" branching, whereas fig. 3-2b shows "phugoid" branching. If one were to compare these loci with those for the basic RPV plant with no actuators (i.e., ideal actuators) (see figs. 3-3a,b), the subtle degrading effects of the actuators are evidenced. Suffice it to say that the range of gains available for improved damping and stability performance are severely restricted, not so much for the "phugoid", but primarily for the "short-period".

Figs. 3-2a,b also show the roots of the closed-loop transfer function for the feedback gain ($K_{\dot{\theta}}^{*}$ = .075) calculated in Appendix B. As stated earlier, $K_{\dot{\theta}}^{*}$ comes directly from the analog A/P gains and is required for analysis. Using Eq (3-1) and $K_{\dot{\theta}}^{*}$, the inner loop is closed to produce the closed-loop transfer function (CLTF) by the expression:

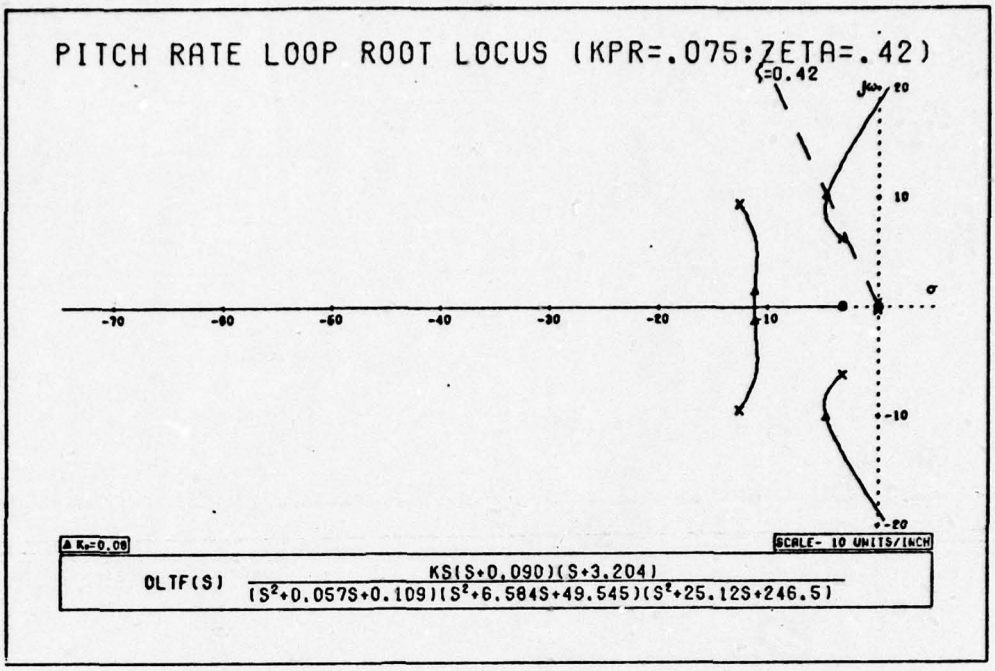


Figure 3-2a: Root Locus of Linear Pitch Rate Controller

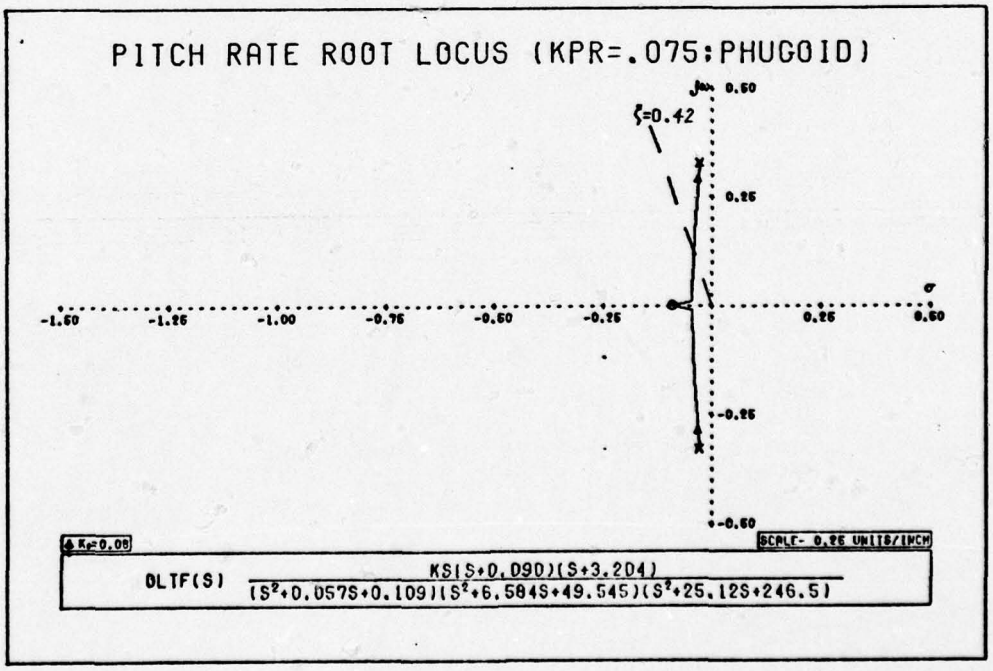


Figure 3-2b: Root Locus of Linear Pitch Rate Controller; Phugoid

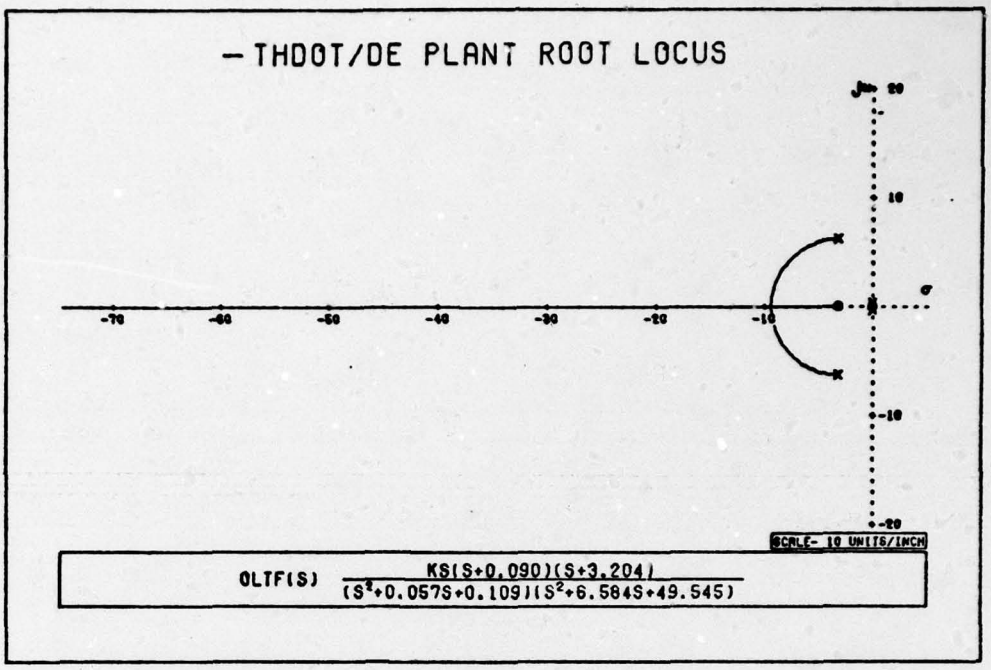


Figure 3-3a: Root Locus of Pitch Rate Transfer Function; $\dot{\theta}(S)/\delta_e(S)$

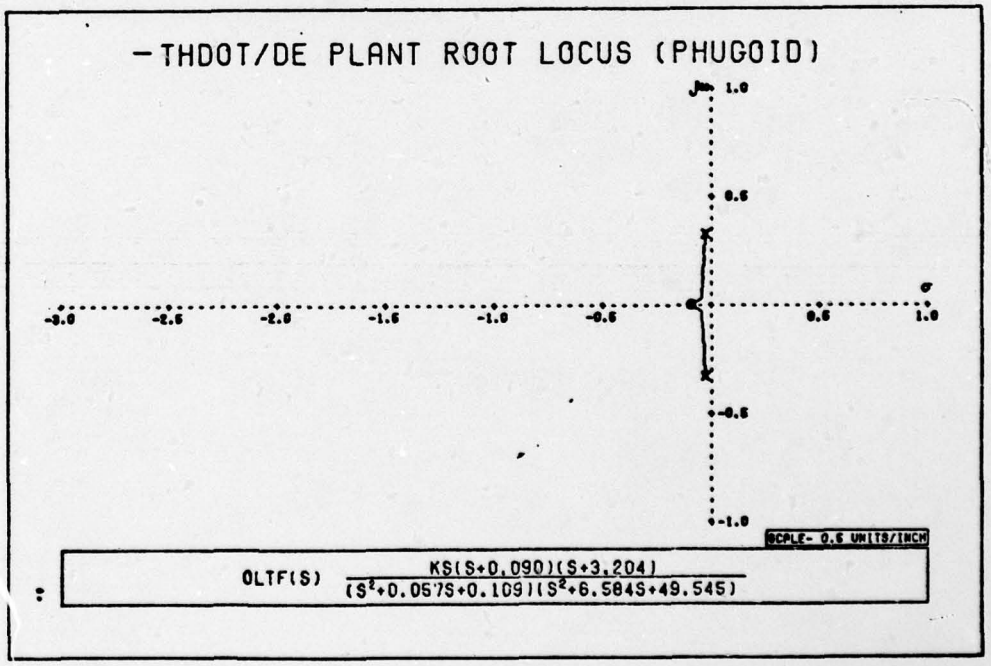


Figure 3-3b: Root Locus of Pitch Rate Transfer Function; Phugoid; $\dot{\theta}(S)/\delta_e(S)$

$$CLTF = \frac{OLTF}{1 + OLTF\ (K_{\dot{\theta}})}$$

This results in the control ratio

$$\frac{\dot{\theta}\ (S)}{\dot{\theta}_c(S)} = \frac{(14809.47)(S+0.090)(S+3.204)(S)}{(S^2+0.063S+0.084)(S^2+9.423S+125.081)(S^2+22.275S+126.29)} \quad (3\text{-}2)$$

This is a stable system since no roots lie in the right-half plane (Ref 15). The closed-loop pitch rate system does not, however, improve the system significantly. A comparison of the time responses and figures of merit (see Table 3-1) for the open-loop ($\dot{\theta}(S)/\delta_{e_c}(S)$; fig. 3-4a,b) versus closed-loop ($\dot{\theta}(S)/\dot{\theta}_c(S)$; fig. 3-5a,b) system demonstrates marginal improvement. In particular, peak time (T_p) decreases by 16.6%, settling time (T_s) decreases by 13.2%, and peak value (P_v) decreases by 2.1%. A very close look at the root locus structure tells why these improvements may be the best attainable by gain variations alone. Better results would have been expected. Also note that the steady-state responses to the unit step go to zero. This is because of the S in the numerator of Eq (3-2) which acts like a differentiator (Ref 4). A frequency response plot is included for reference in fig. 3-6.

Verification of the pitch rate loop model compared to the analog A/P cannot be made at this time because time responses for the loop were not made via simulation. Closing the outer attitude loop and comparing recorded responses will, however, provide model validity.

Table 3-1

Figures of Merit: Longitudinal Controllers

Transfer Function	T_R	T_D	T_P	T_S	P_V	F_V	Comments
$\dot{\theta}(S)/\delta_e(S)$	Undefined	Undefined	.251	119.920	7.225	0	Model
$\dot{\theta}(S)/\delta_{e_c}(S)$	Undefined	Undefined	.370	120.022	6.994	0	Model
$\dot{\theta}(S)/\dot{\theta}_c(S)$	Undefined	Undefined	.308	104.182	5.479	0	Model, $K_{\dot{\theta}} = .075$
$\theta(S)/\theta_c(S)$	.284	.517	2.960	----	19.955	12.007	Model, $K_{\theta_c} = .39$ Step = 20 deg
$\theta(S)/\theta_c(S)$	$\approx .3$	.5	2.9	----	20.	12.	Simulation, $K_{\theta_c} = .34$ Step = 20 deg

T_R = Rise time, sec

T_D = Duplication time, sec

T_P = Peak time, sec

T_S = Settling time, sec

P_V = Peak value

F_V = Final value

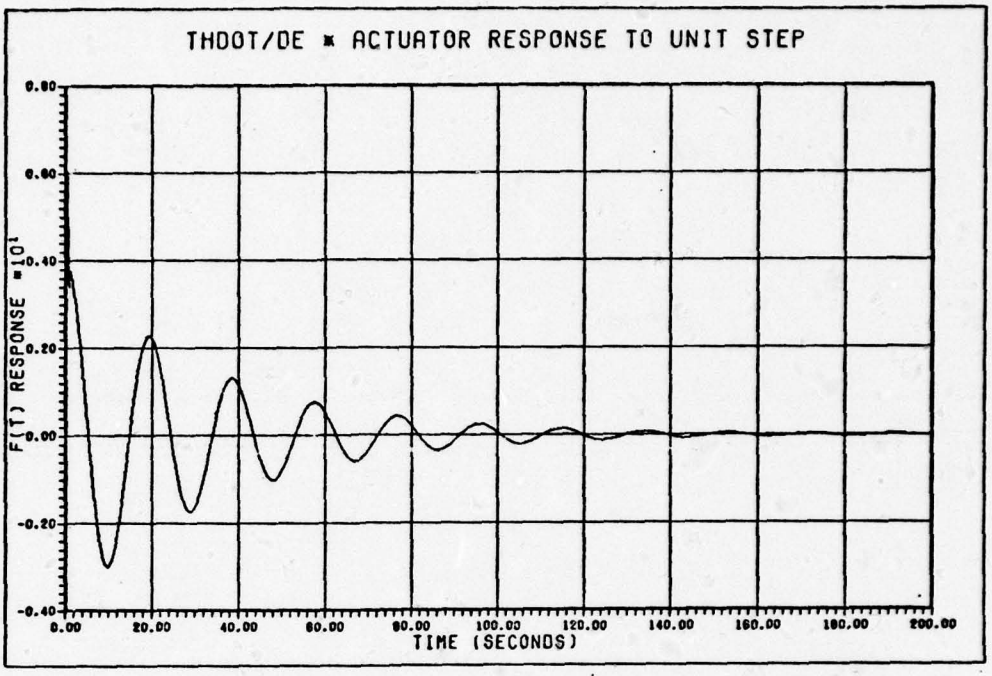


Figure 3-4a: Time Response of $\dot{\theta}(S)/\delta_{e_c}(S)$; Unit Step

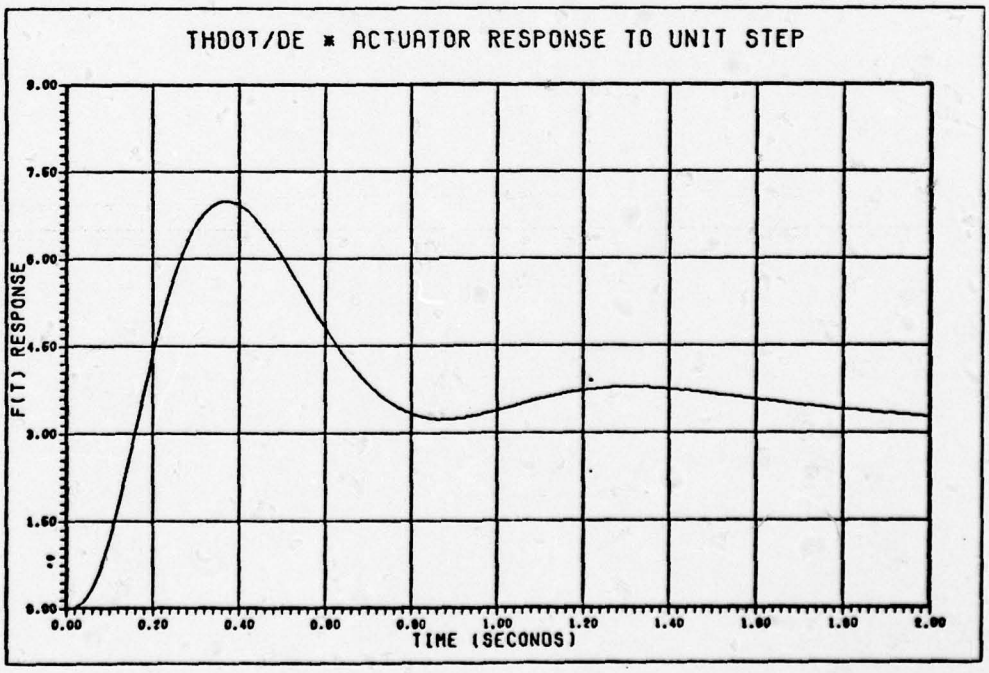


Figure 3-4b: Time Response of $\dot{\theta}(S)/\delta_{e_c}(S)$; Unit Step; Phugoid

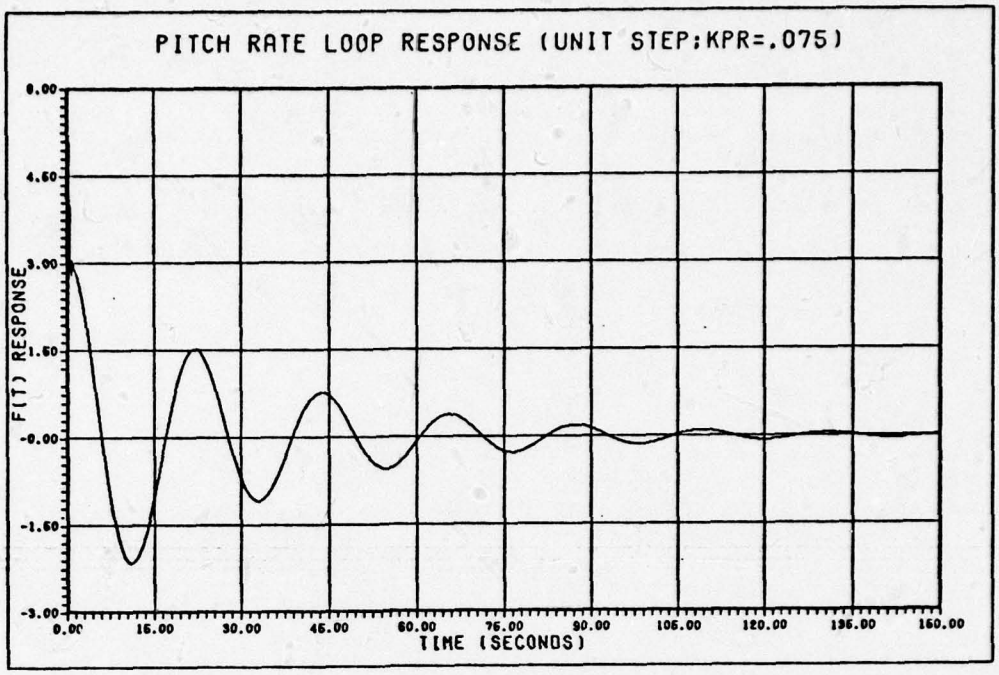


Figure 3-5a: Time Response of $\dot{\theta}(S)/\dot{\theta}_c(S)$; Unit Step

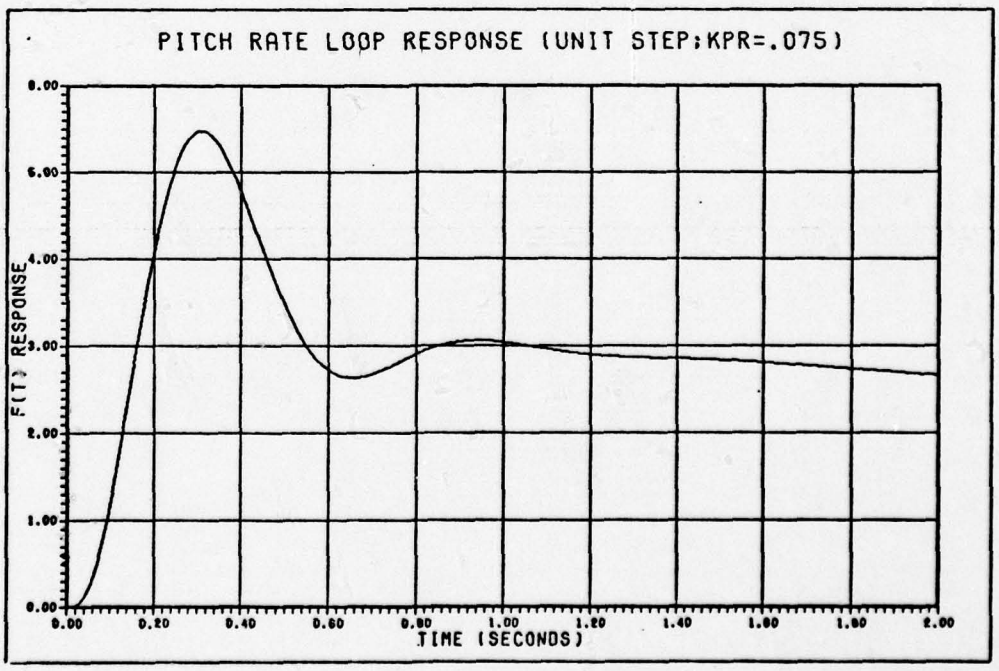


Figure 3-5b: Time Response of $\dot{\theta}(S)/\dot{\theta}_c(S)$; Unit Step; Phugoid

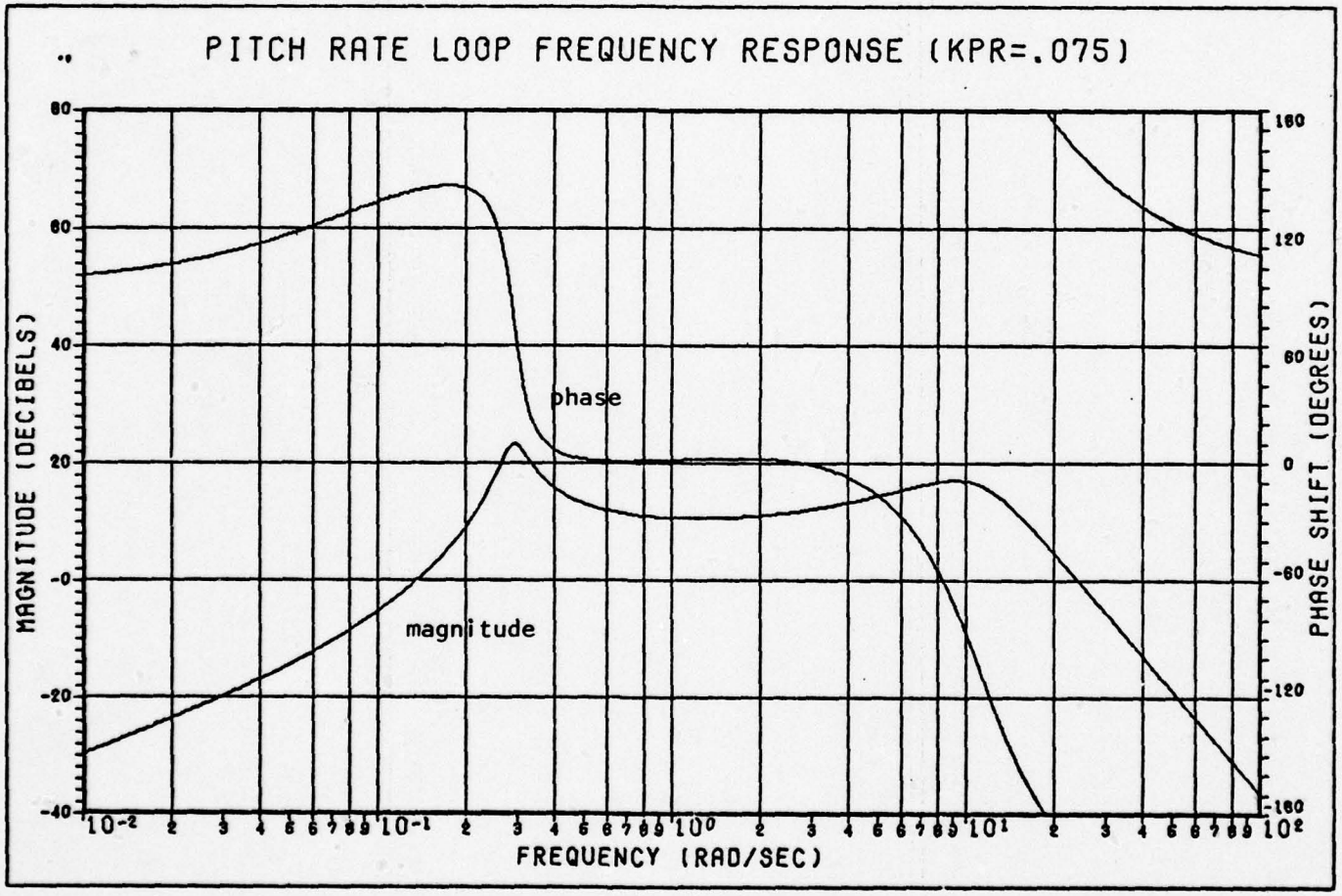


Figure 3-6: Frequency Response of Linear Pitch Rate Controller

Linear Pitch Attitude Controller. The linear pitch attitude controller is formed by closing the attitude (outer) loop around the closed pitch rate (inner) loop (see fig. 3-7 for block diagram). Pitch attitude is detected by a vertical gyro whose transfer function is unity (see chapter II). The signal is fed back, multiplied by a suitable feedback gain (K_θ, KPA) and subtracted from the attitude pitch command (θ_c, THCMD). The resulting error signal becomes a pitch rate command (e_θ) to the inner loop.

Manipulation of the model yields the new open-loop transfer function:

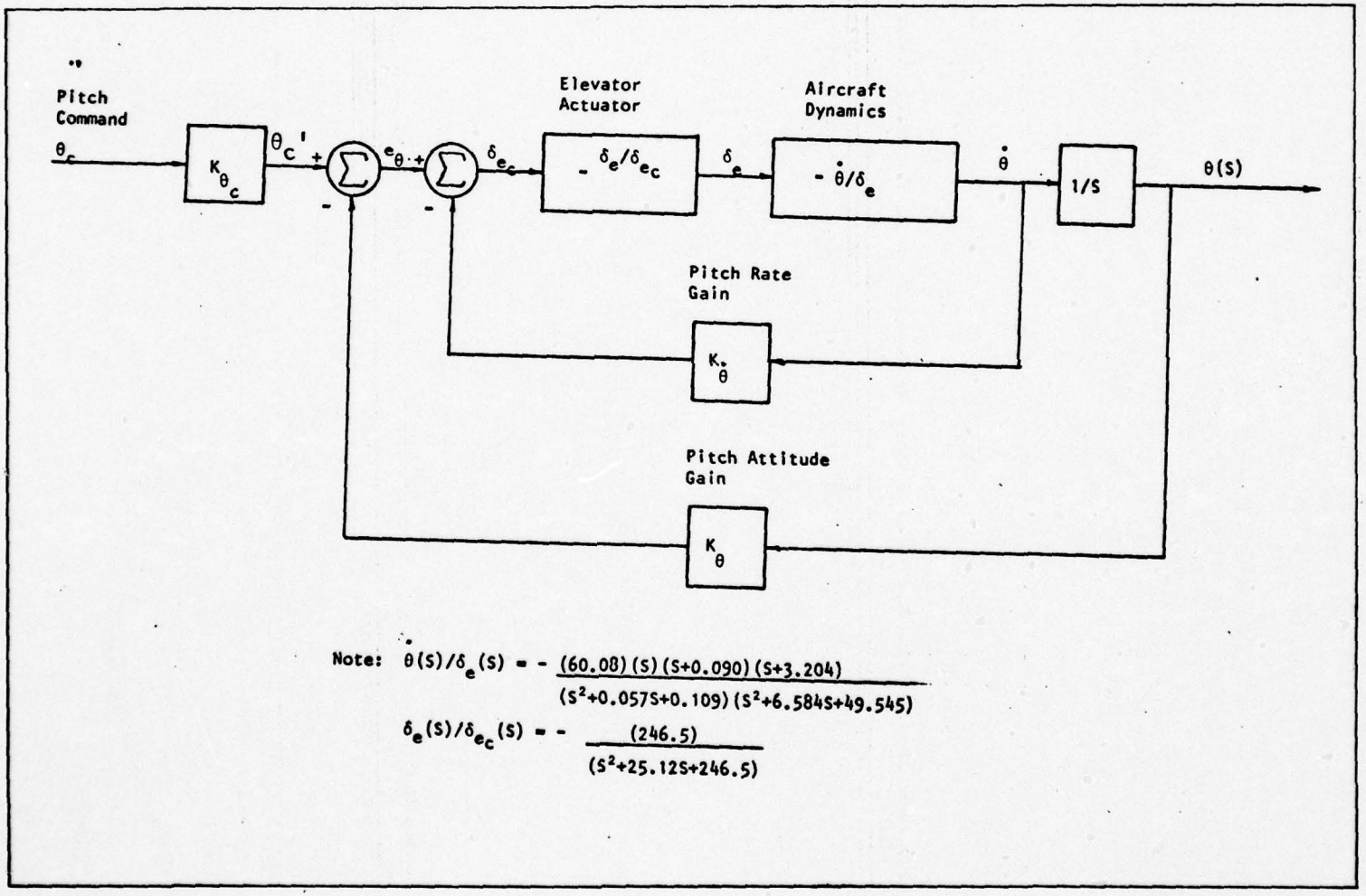
$$\frac{\theta(S)}{e_\theta(S)} = \frac{(14809.47)(S+0.090)(S+3.204)}{(S^2+0.063S+0.084)(S^2+9.423S+125.081)(S^2+22.275S+126.29)} \quad (3-3)$$

which is a type-0 system. The root loci for Eq (3-3) are shown in figs. 3-8a,b.

Using the necessary feedback gain, $K_\theta = 0.34$, (derived in Appendix B), and $K_{\theta_c} = 1.0$, the outer loop is closed to form the control ratio:

$$\frac{\theta(S)}{\theta_c(S)} = \frac{(14809.47)(S+0.090)(S+3.204)}{(S+0.789)(S+0.202)(S^2+6.516S+100.617)(S^2+24.26S+174.125)} \quad (3-4)$$

Inspection of the roots of Eq (3-4) reveals a stable type-0 system. Because of this, one would expect a steady-state error to exist (Ref 15:178). Application of the final value theorem to Eq (3-4) using the full equation coefficients from TOTAL and a step unit, $\theta_c(t) = R_o u_{-1}(t)$, yields (Ref 15: 103)



Note: $\dot{\theta}(S)/\delta_e(S) = -\dfrac{(60.08)(S)(S+0.090)(S+3.204)}{(S^2+0.057S+0.109)(S^2+6.584S+49.545)}$

$\delta_e(S)/\delta_{e_c}(S) = -\dfrac{(246.5)}{(S^2+25.12S+246.5)}$

Figure 3-7: Block Diagram of Linear Pitch Attitude Controller

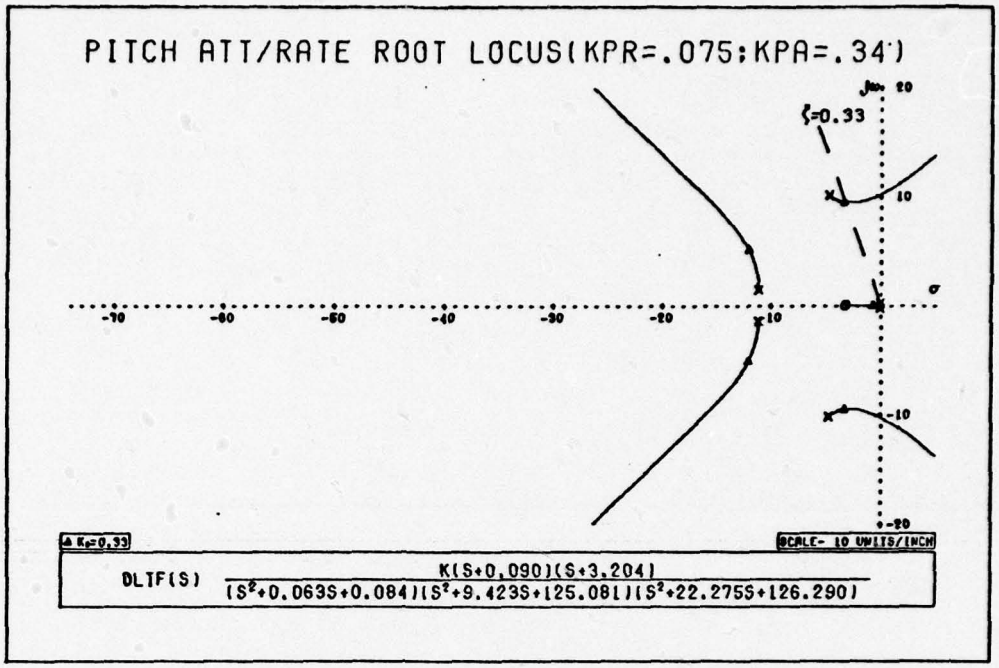


Figure 3-8a: Root Locus of Linear Pitch Attitude Controller

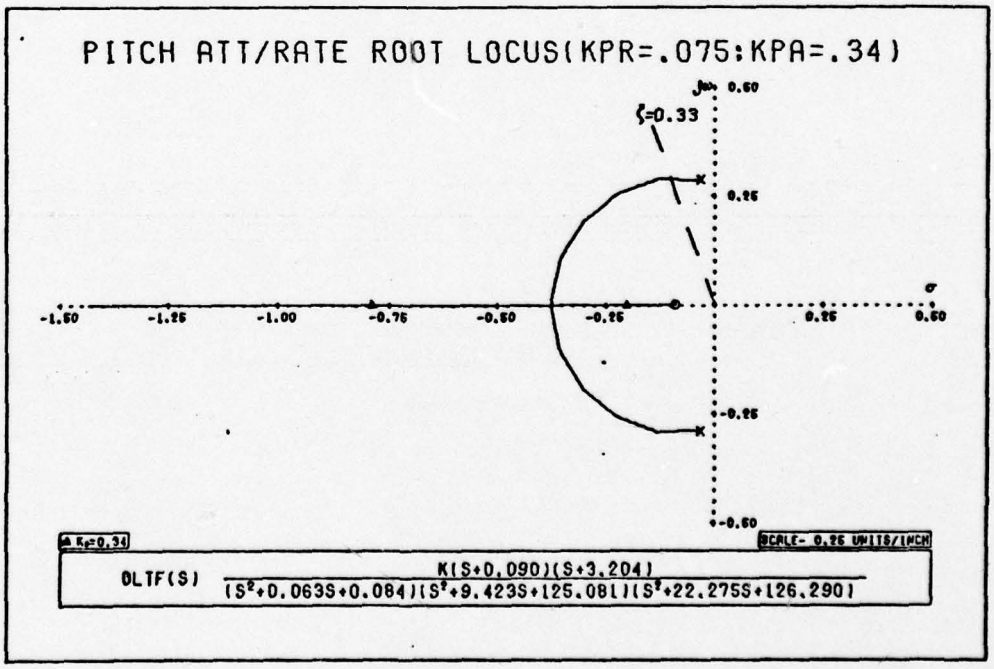


Figure 3-8b: Root Locus of Linear Pitch Attitude Controller; Phugoid

$$\theta(t)_{ss} = \lim_{S \to 0} S\theta(S) = 1.539R_o \qquad (3\text{-}4a)$$

Eq (3-4a) clearly shows the expected steady-state output to be 1.539 times larger than the input.. If, for example, a 20 degree pitch up is commanded, the corresponding output attitude will be 30.78 degrees. This is obviously undesirable and attenuation is needed. A solution exists, however, to remedy this condition. From fig. 3-7, and $K_{\theta_c} \neq 1$, the closed-loop transfer function is

$$\frac{\theta(S)}{\theta_c(S)} = K_{\theta_c}\left[\frac{G(S)}{1 + G(S)K_\theta}\right] \qquad (3\text{-}5)$$

where G(S) = Eq (3-3).

Taking the final value of Eq (3-5), for an arbitrary step input, $\theta_c(t) = R_o u_{-1}(t)$, yields

$$\theta_{ss}(t) = \lim_{S \to 0}\left[SK_{\theta_c}\left[\frac{G(S)}{1 + G(S)K_\theta}\frac{R_o}{S}\right]\right] = 1.539\ K_{\theta_c}\ R_o \qquad (3\text{-}6)$$

Thus, in order for output attitude to equal input attitude, $\theta_{ss}(t) = R_o = \theta_c$, then

$$K_{\theta_c} = \frac{1}{1.539} = 0.65 \qquad (3\text{-}7)$$

This substantiates the assumption of needing an attenuator to eliminate the steady-state error.

At this point, a frequency response for Eq (3-5), with $K_{\theta_c} = 1$, is shown in fig. 3-9a. Inspection of the magnitude plot reveals a hump prior to rolloff. This indicates second-order dynamics which, in turn, implies the possible existence of a time-response overshoot peak. If this fact is true, then the degree/magnitude of overshoot is crucial to the attitude controller response. If given a 20 degree pitch up command, the RPV may stall as it pitches up greater than 20 degrees.

As a result, the peak value figure of merit calculated by TOTAL for Eq (3-4), assuming $K_{\theta_c} = 1$ and a unit step, is

$$P_V = 2.55821 \tag{3-8}$$

Letting Eq (3-8) represent the new interim final value for any step input, $R_o\, u_{-1}(t)$, and $K_{\theta_c} \neq 1$, yields

$$\theta_{ss}(t) = 2.55821\, K_{\theta_c} R_o \tag{3-9}$$

Requiring $\theta_{ss}(t) = R_o = \theta_c$ and solving Eq (3-9) for K_{θ_c} produces

$$K_{\theta_c} = \frac{1}{2.55821} = 0.391 \tag{3-10}$$

A frequency response of Eq (3-4) with $K_{\theta_c} = 0.391$ is shown in fig. 3-9b. Controller bandwidth is approximately 12 rad/sec (1.9 HZ).

Time responses of Eq (3-4) with $K_{\theta_c} = 0.39$ for a 20 degree pitch up command ($\theta_c = 20$ deg) are depicted in figs. 3-10a,b. Note that the

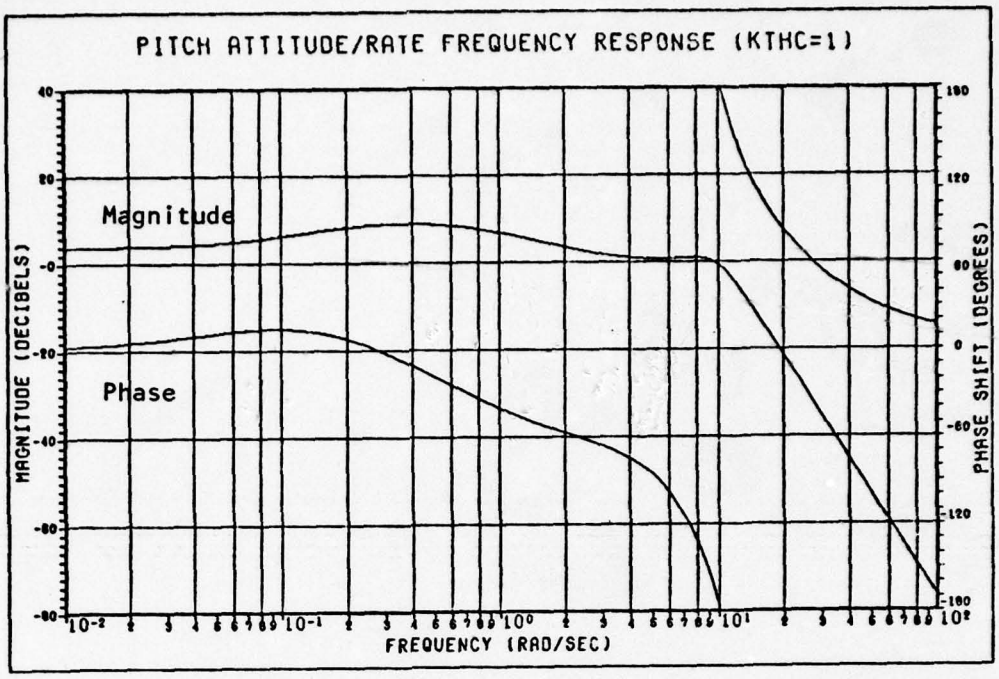


Figure 3-9 a: Frequency Response of Linear Pitch Attitude Controller; $K_{\theta_c} = 1$

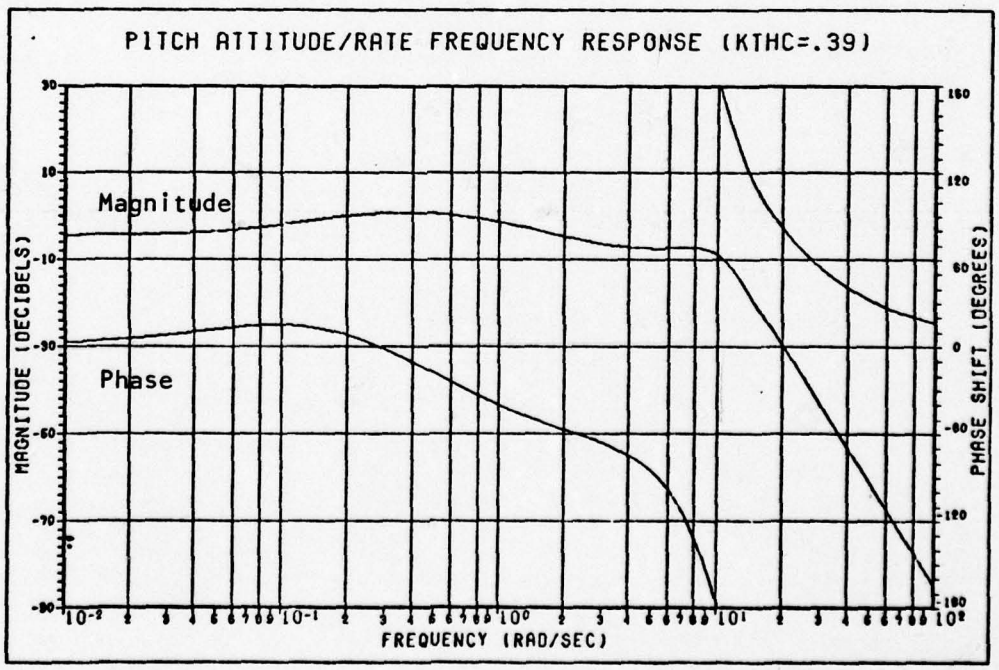


Figure 3-9 b: Frequency Response of Linear Pitch Attitude Controller; $K_{\theta_c} = .39$

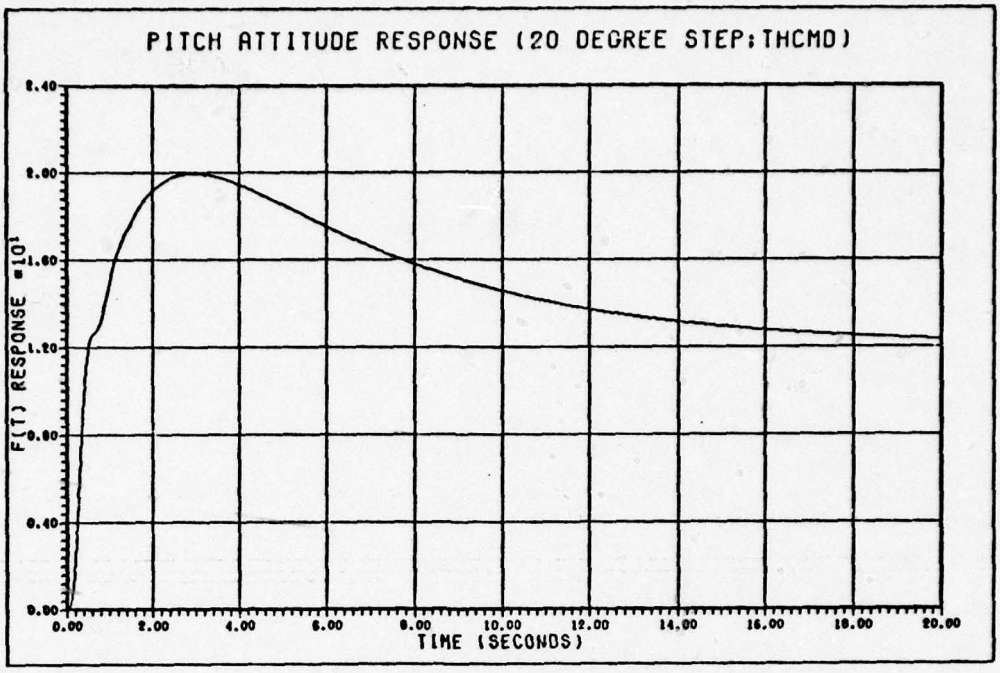


Figure 3-10a: Time Response of Linear Pitch Attitude Controller; 20 sec ; $K_{\theta_c} = 0.39$

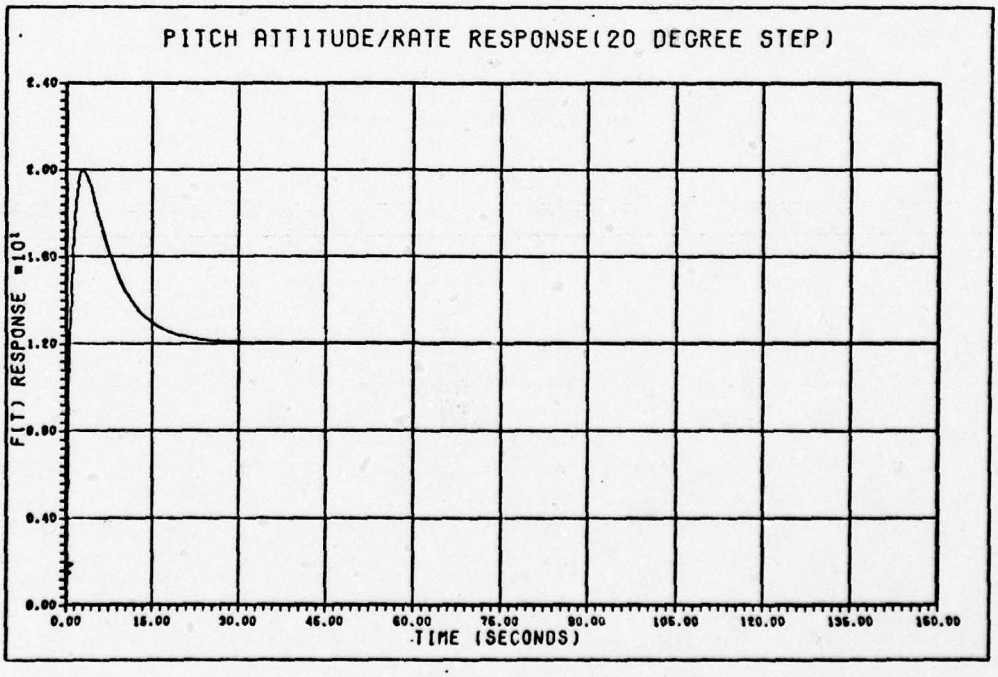


Figure 3-10b: Time Response of Linear Pitch Attitude Controller; 150 sec ; $K_{\theta_c} = 0.39$

commanded 20 deg attitude is achieved at peak with a resultant 12 degree steady-state attitude. These results, along with the figures of merit in table 3-1, confirm the theoretical model development.

The time response exhibited in fig. 3-10a,b is a "standard" pitch attitude controller response and is well documented (Ref 4:79). Subject to an input step command, the RPV pitches to the commanded attitude, then slowly glides/descends to a lower steady-state attitude. Pitch attitude controllers alone cannot correct this aerodynamic property. Some form of additional velocity control is needed to obtain the commanded steady-state attitude (Ref 4:79).

Having designed a linear pitch attitude controller model, the next step is to compare it with the analog controller simulation. Fig. 3-11 illustrates analog controller responses to the same 20 degree pitch up command (θ_c) forcing function used in the model. In order to reproduce, as closely as possible, the model response shown in fig. 3-10a, the gain, K_{θ_c}, on the analog controller simulation, was adjusted to 0.34. This represents an error of only 12%. Figures of merit (see table 3-1) show a good deal of similarity between the two responses (similar T_R, T_D, T_P, P_V, F_V) and provide a modest amount of confidence in the modelling process. Thus, it is shown clearly, and within an acceptable margin of linearization error, that the analog pitch attitude controller can be modelled accurately with linear controller.

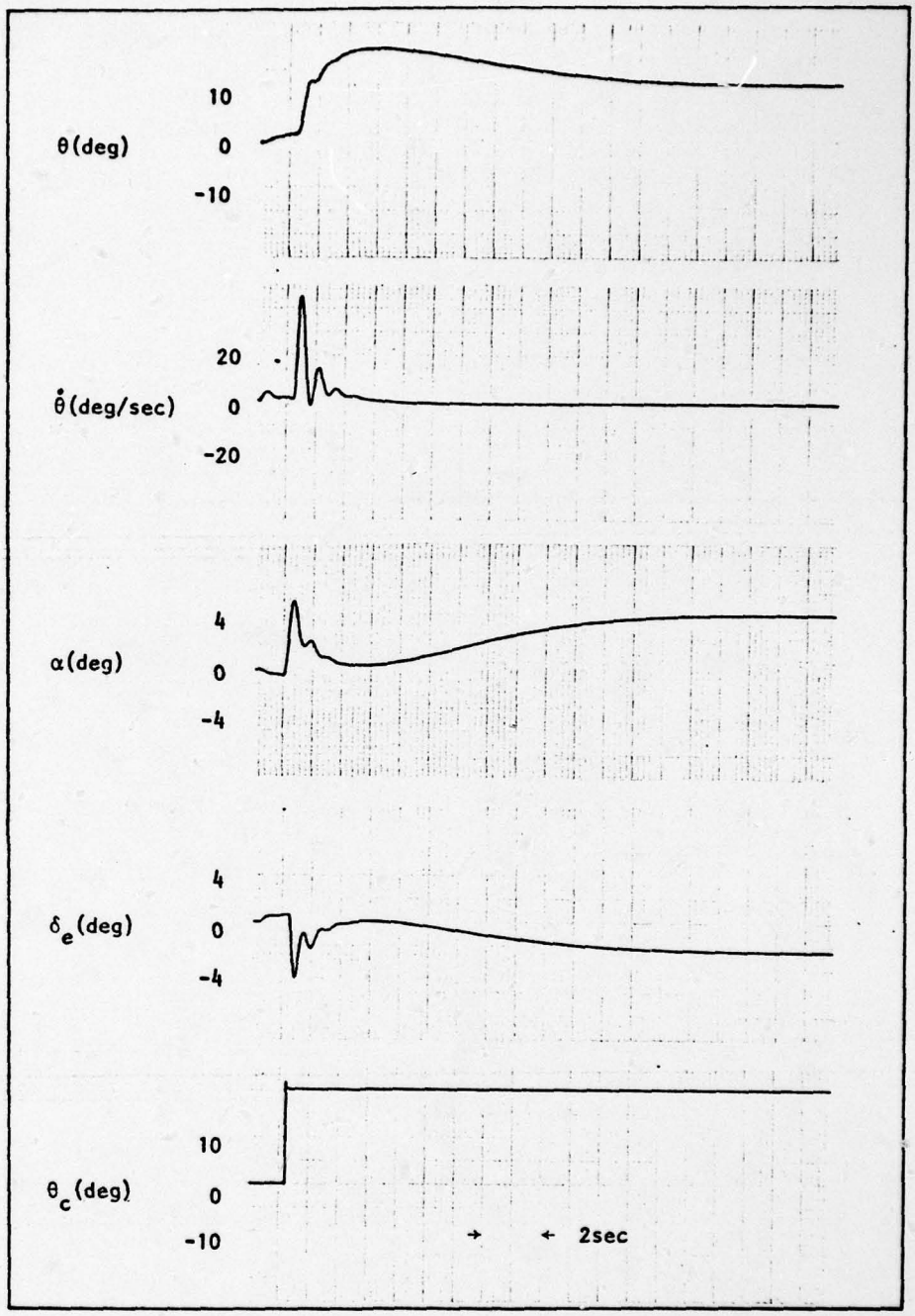


Figure 3-11: RPV Response to 20 Degree Step Pitch Up Command(θ_c); Pitch Attitude/Rate Loops Closed; Analog Controller Engaged

Recommendations. Without going into detail, the root loci for both the pitch rate and pitch attitude loops reveal that system performance may not be further improved by loop gain variations alone. If improved performance is required, the current complex-pole actuators may have to be replaced with ones of higher frequency dynamics and real poles, or lead compensation placed in the feedback loops. This will enable larger loop gains which, in turn, means greater controller bandwidth (i.e., BW > 1.9 HZ) and "muscle power".

Lateral-Directional Controllers

The usefulness of investigating feedback loops closed around the various RPV transfer functions is exactly the same in the case of lateral motions as it is for longitudinal. This section will develop the controllers for both the roll and yaw axes that have been illustrated in fig. 2-5, chapter II. Again, as in the longitudinal case, it is sufficient to consider each controller as a simple gain which has already been determined (see Appendix B).

Typically, three dynamic characteristics govern lateral-directional flight: spiral divergence, roll subsidence and Dutch roll (Ref 32:459). Spiral divergence motion consists of a combination of increase in yaw angle and roll angle with resulting high speed spiral dive. The time constant is usually very large; τ_{S_D} = 812 sec for Case II. Since the RPV exhibits "slight" divergence, i.e., pole is in right half plane (S=.001231), the purpose of using an A/P is to impose a higher degree of stability to this mode. Roll subsidence, on the other hand, is the rolling response of the aircraft to an aileron input and is described for Case II as having a time constant equal to .116 sec (τ_{R_S} = .116 sec for S = -8.6). Dutch roll, primarily excited by the rudder (Ref 4:132), consists of both roll and yaw angle excursions. It is usually lightly damped (ζ = .266, ω_n = 48.1 rad/sec for Case II) and may become objectionable. There is a need to control this mode through yaw rate damping (Ref 29). Rudder deflections primarily excite the Dutch roll mode, whereas those for the aileron excite roll rate and roll angle with some Dutch roll superimposed in this motion (Ref 4:138).

Linear Roll Rate Controller. The fundamental block diagram for the linear roll rate controller is shown in fig. 3-12. Control of roll rate is provided for by feeding back roll rate detected by a unity gain rate sensor, multiplying the signal by a gain ($K_{\dot{\phi}}$, KRR) and subtracting it from the roll rate command ($\dot{\phi}_c$). The error signal thus produced drives the aileron actuator to the required position. Transient and steady-state characteristics are determined by the feedback gain ($K_{\dot{\phi}}$).

From Appendix A the RPV transfer function (also shown in fig. 3-12), relating roll rate ($\dot{\phi}$) to aileron deflection (δ_a) is found to be:

$$\frac{\dot{\phi}(S)}{\delta_a(S)} = \frac{(112.5)(S^2+2.477S+22.169)(S)}{(S-.001)(S+8.606)(S^2+2.559S+23.146)} \qquad (3\text{-}11)$$

When Eq (3-11) is multiplied by the aileron actuator transfer function, the open-loop transfer function becomes

$$\frac{\dot{\phi}(S)}{\delta_{a_c}(S)} = \frac{(27730)(S^2+2.477S+22.169)(S)}{(S-.001)(S+8.606)(S^2+2.559S+23.146)(S^2+25.12S+246.5)} \qquad (3\text{-}12)$$

which is a type-0 system. The root locus of Eq (3-12), depicted in fig. 3-13a, shows that the linear roll rate controller system is completely unstable for all values of gain. This is primarily attributed to the spiral divergence root (S = .001) in Eq (3-12) which lies in the right-half plane. Also note that the Dutch roll mode is almost eliminated due to the numerator zeros cancelling the denominator poles. Thus, and as expected, roll rate inputs do not excite Dutch roll oscillations. In

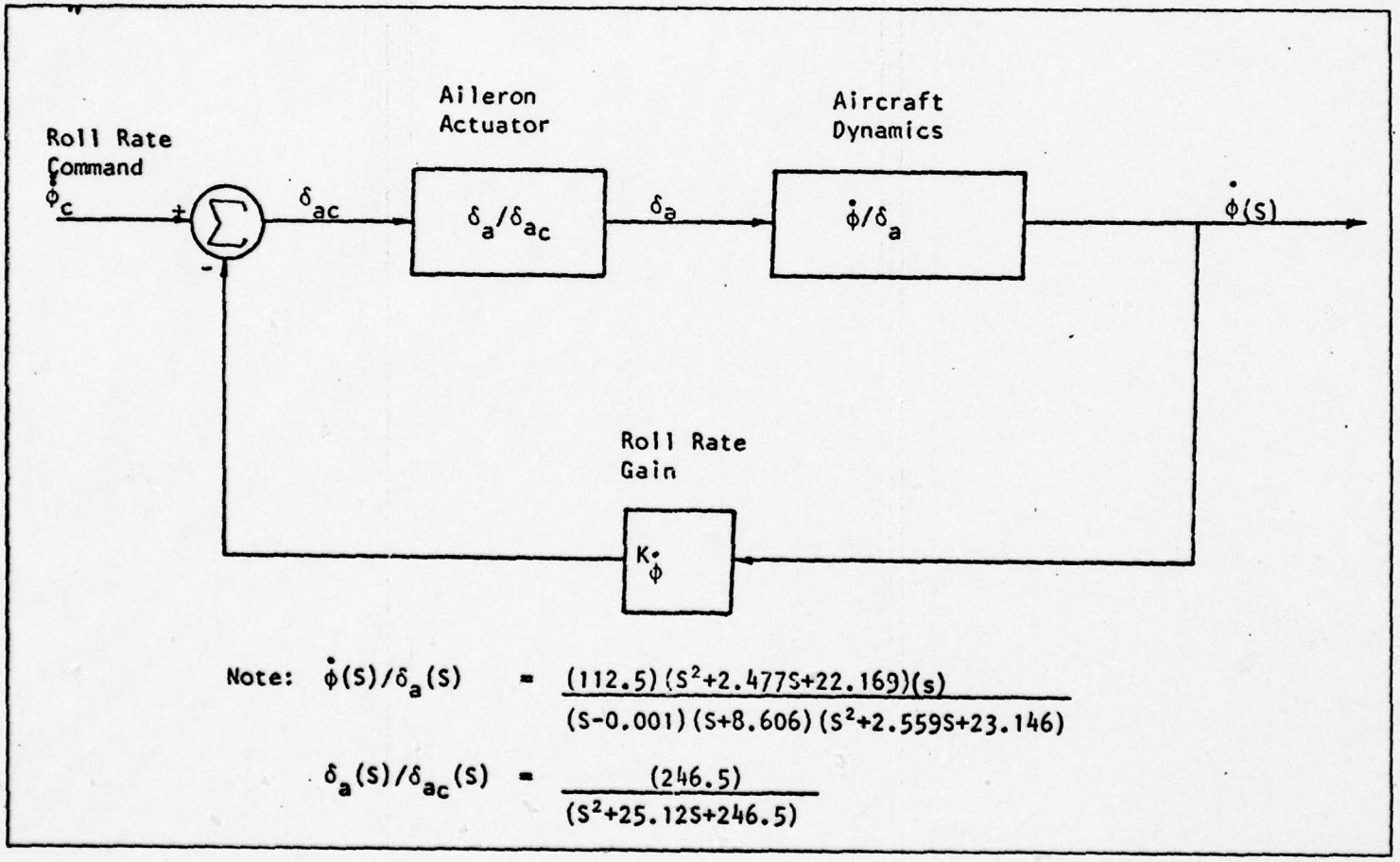


Figure 3-12: Block Diagram of Linear Roll Rate Controller

addition to the "slightly" unstable spiral mode, the complex actuator poles (see fig. 3-13a) further degrade the controller root locus for variations of gain. A comparison of this locus with that of the basic $\dot{\phi}(S)/\delta_a(S)$ plant (see fig.3-13b), without an actuator, clearly shows the difference. At first glance, this discussion of degrading an already unstable system seems meaningless. However, when the outer attitude loop is closed, in the next section, this rationale will prove noteworthy.

The closed-loop transfer function for $K_{\dot{\phi}}^{\cdot} = 0.116$ (extracted from Appendix B) is

$$\frac{\dot{\phi}(S)}{\dot{\phi}_c(S)} = \frac{(27730)(S)(S^2+2.477S+22.171)}{(S-.0005)(S+23.718)(S^2+2.554S+22.461)(S^2+10.012S+226.030)} \quad (3\text{-}13)$$

which is a type-0 system. Fig. 3-13a indicates the pole locations of Eq (3-13) for a damping ratio of 0.33 ($\zeta = .33$). Note that Eq (3-13) also represents an unstable system due to the spiral divergence term ($S = .0005$). The corresponding time constant is very large (33 min). Thus, this system can be said to exhibit a "slight long-term" instability characteristic which presents no problem to the control analysis system handling qualities. A roll rate frequency response is included in fig. 3-14 to show a system bandwidth of about 30 rad/sec (4.77 HZ).

Since no hybrid simulation for roll rate was recorded, a comparison of the model with the analog rate controller is not addressed. It is appropriate, however, to show the degrading effects upon performance that closing the roll rate loop produces; particularly interesting is decreased

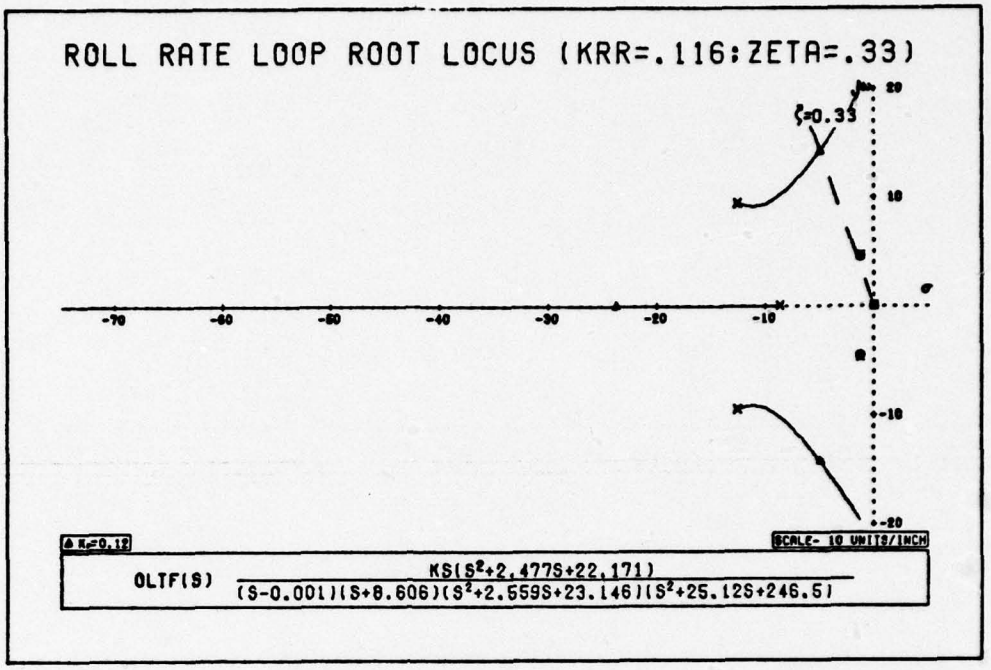


Figure 3-13a: Root Locus of Linear Roll Rate Controller

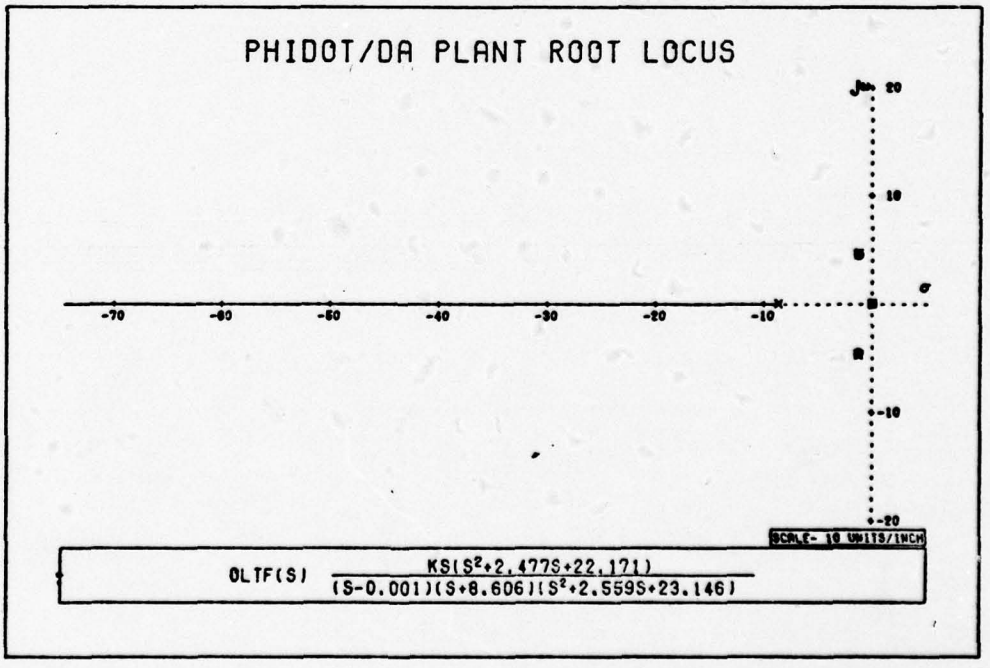


Figure 3-13b: Root Locus of $\dot{\phi}(S)/\delta_a(S)$ Without Actuator

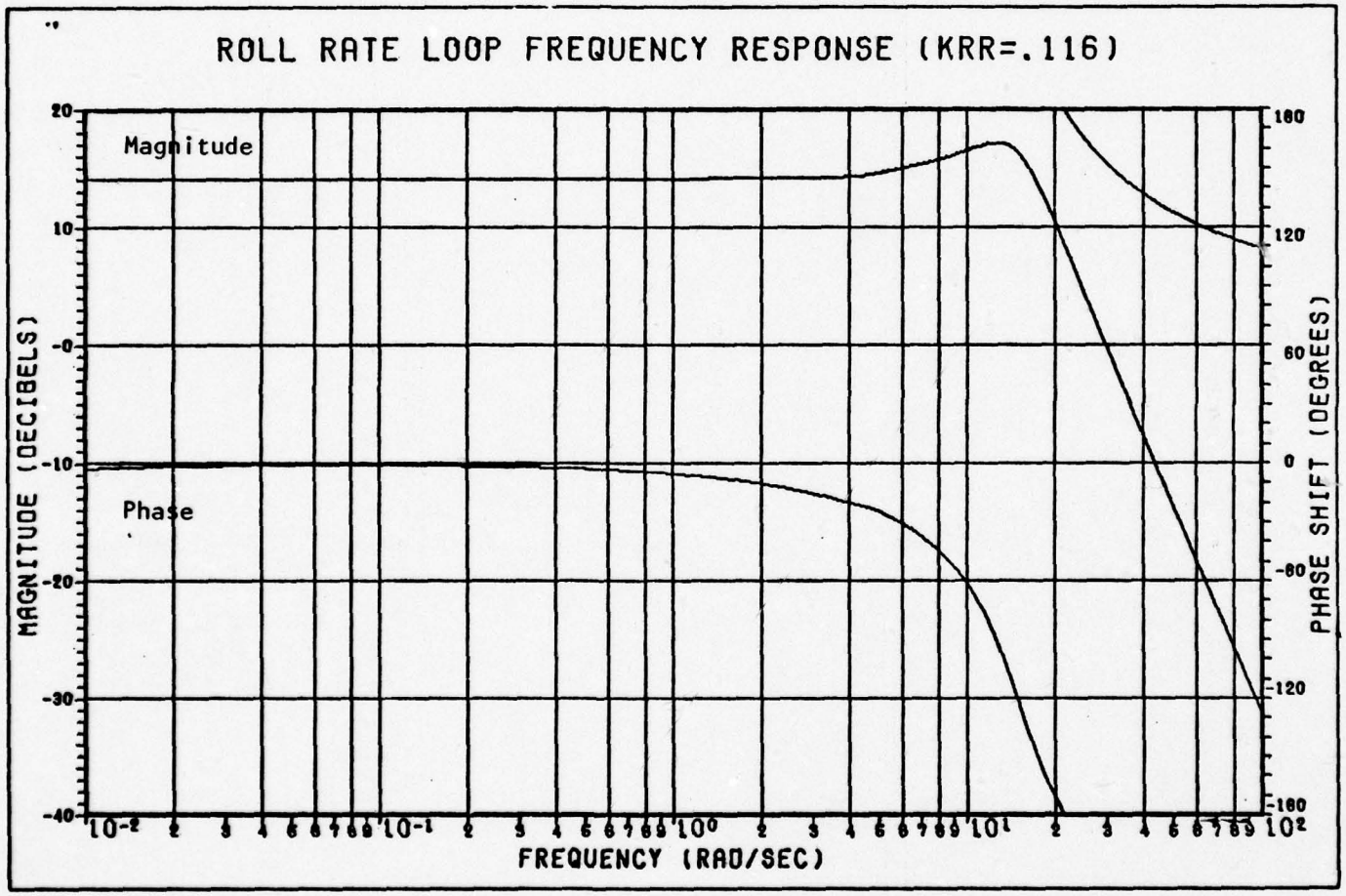


Figure 3-14: Frequency Response of Linear Roll Rate Controller

damping and increased peak magnitude. Compare fig. 3-15a (closed roll rate loop) with fig. 3-15b (open roll rate loop) for a unit pulse of unit width. Fig. 3-16 is also included to illustrate the response of Eq (3-13) to a unit step.

Linear Roll Attitude Controller. The linear roll attitude controller is formed by closing a position loop around the closed-loop roll rate controller (see block diagram fig. 3-17). The roll rate inner loop determines the open-loop poles of the attitude controller. Again, the feedback gain (K_ϕ, KRA) is calculated in Appendix B from the analog A/P. Outer loop operation is similar to that described in the longitudinal controller section.

By integrating Eq (3-13) (i.e., deleting the numerator S) the open-loop transfer function becomes:

$$\frac{\phi(S)}{e_\phi(S)} = \frac{(27730)(S^2+2.477S+22.171)}{(S-.0005)(S+23.718)(S^2+2.554S+22.461)(S^2+10.012S+226.030)} \quad (3\text{-}14)$$

The root locus for this type-0 function is depicted in fig. 3-18.

By incorporating the gains, $K_\phi = 0.267$ and $K_{\phi_c} = 1.0$, calculated in Appendix B, into the analysis and closing the outer loop, the closed loop transfer function becomes

$$\frac{\phi(S)}{\phi_c(S)} = \frac{(27730)(S^2+2.477S+22.171)}{(S+1.556)(S+23.11)(S^2+2.58S+22.376)(S^2+9.036S+203.744)} \quad (3\text{-}15)$$

The poles of this equation for the damping ratio, 0.31, ($\zeta = 0.31$), appear in fig. 3-18 and are shown to be stable (i.e., no roots in right half-

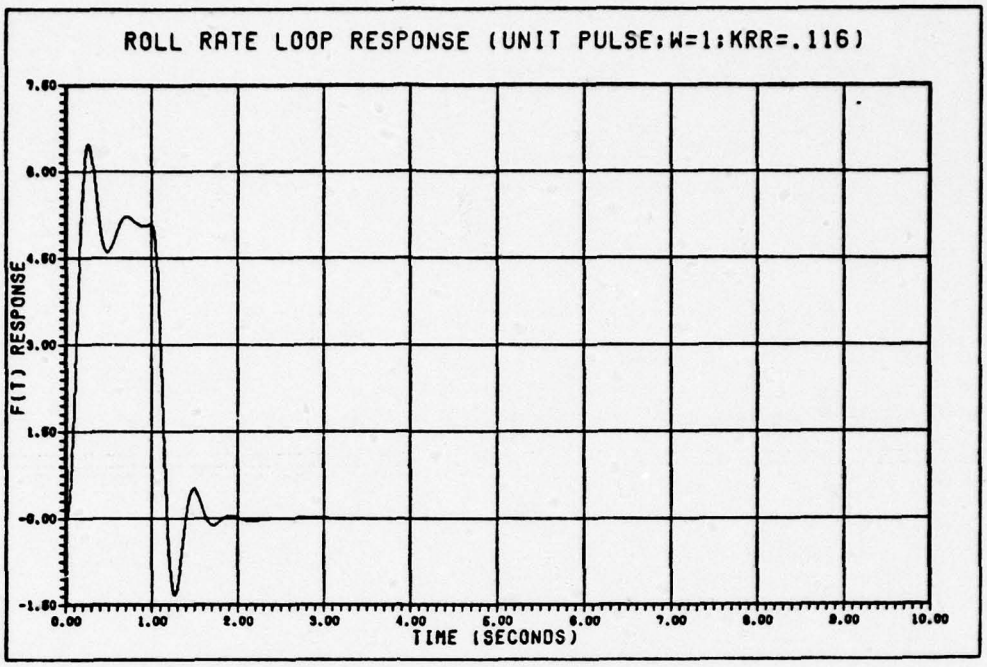


Figure 3-15a: Time Response of Linear Roll Rate Controller; Unit Pulse

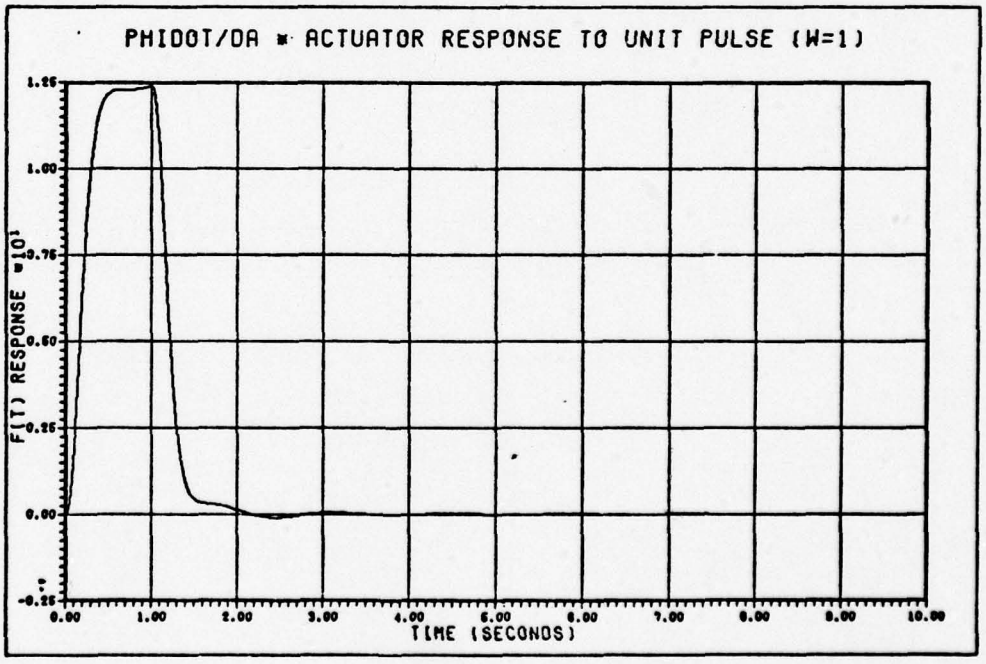


Figure 3-15b: Time Response of $\dot{\phi}(S)/\delta_{a_c}(S)$; Open Loop

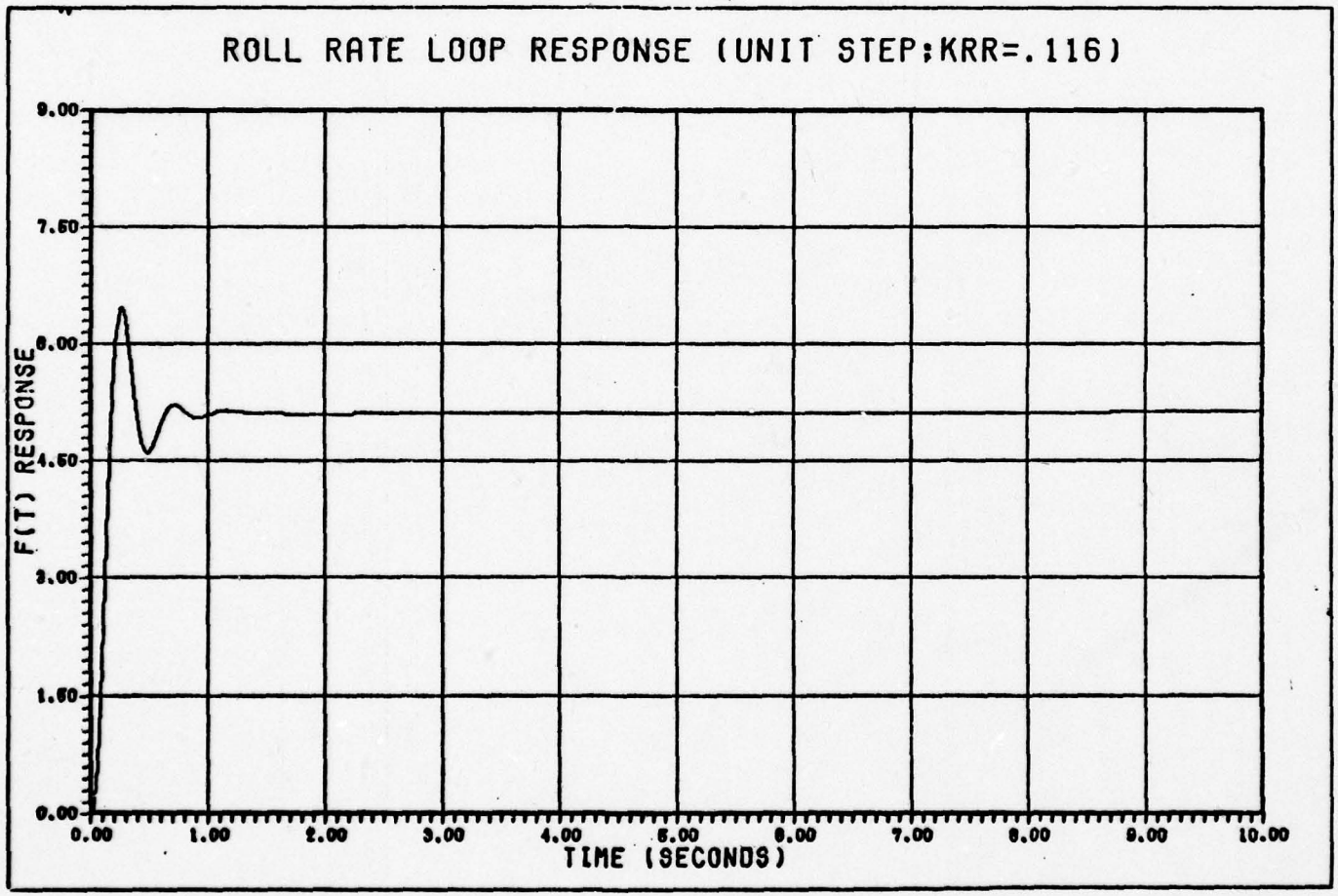


Figure 3-16: Time Response of Linear Roll Rate Controller; Unit Step

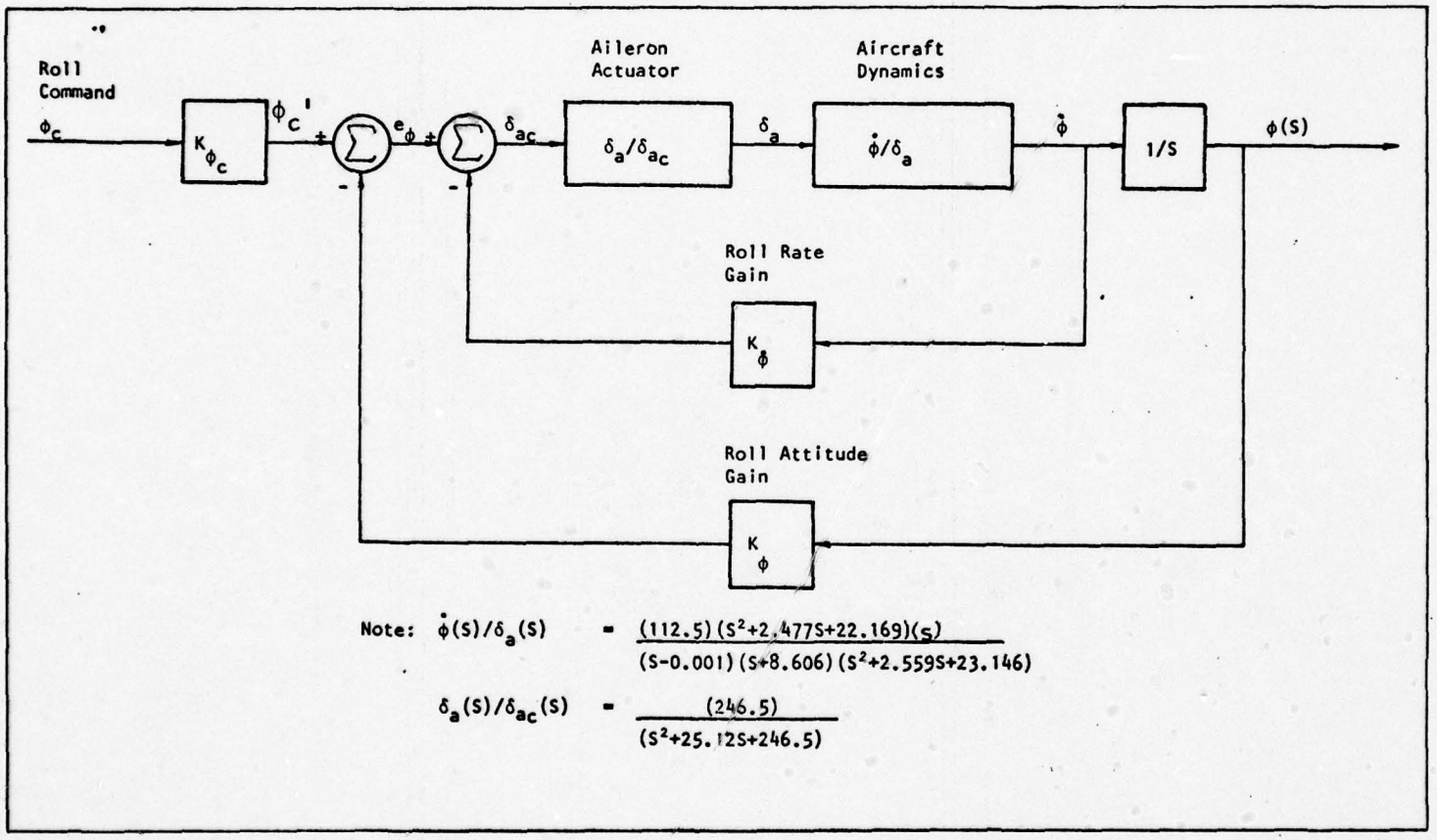


Figure 3-17: Block Diagram of Linear Roll Attitude Controller

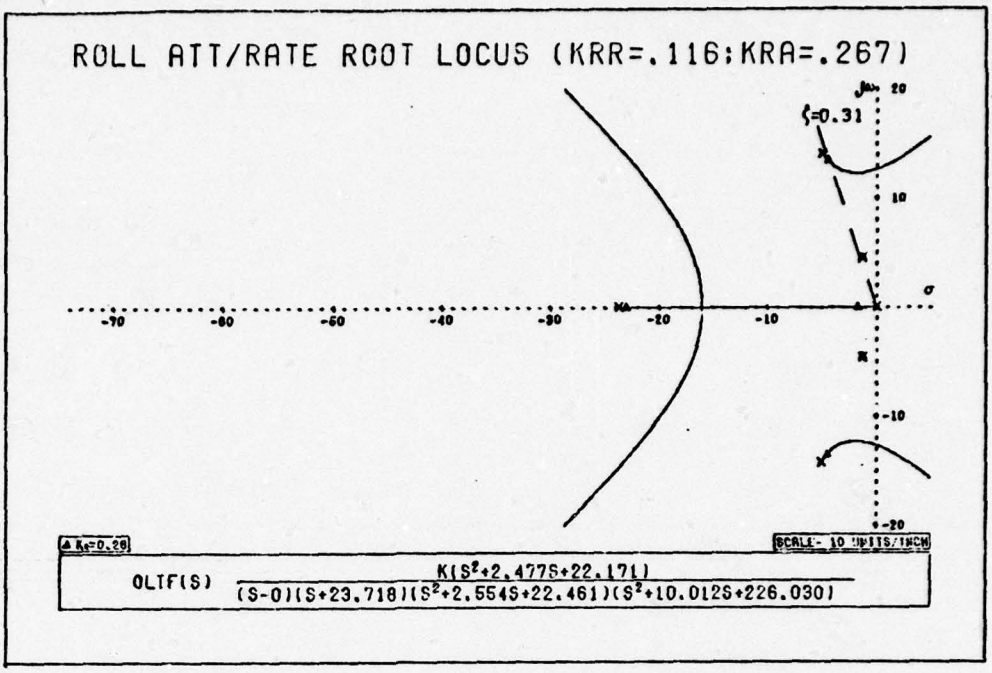


Figure 3-18: Root Locus of Roll Attitude Controller

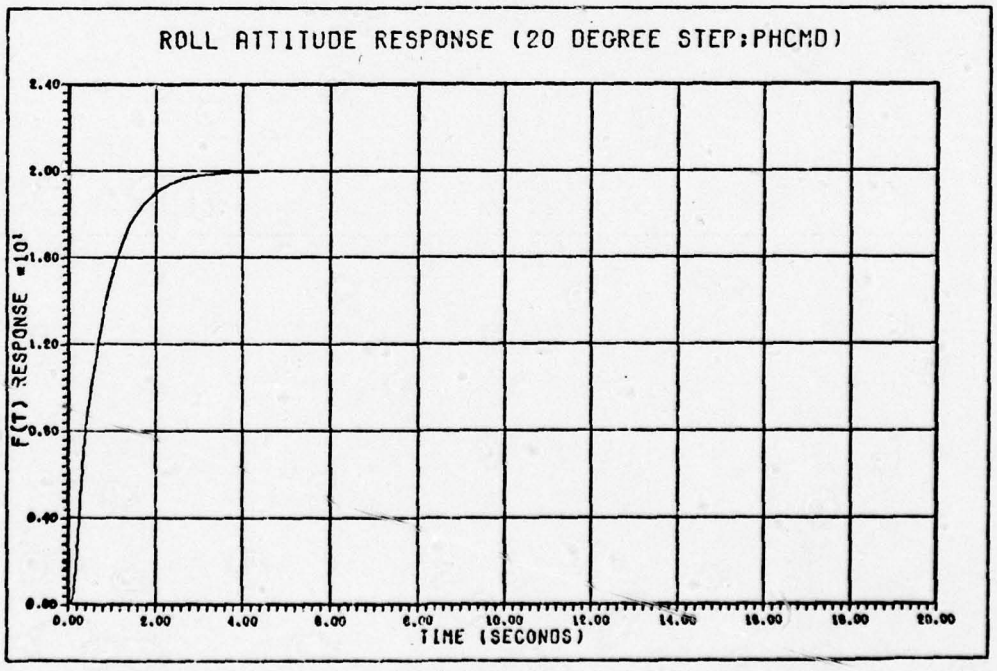


Figure 3-20: Time Response of Roll Attitude Controller

plane). For an arbitrary step input, $\phi_c(t) = R_o u_{-1}(t)$, and $K_{\phi_c} = 1.0$, the final value of Eq (3-15), as determined by TOTAL, becomes

$$\phi_{ss}(t) = \lim_{S \to 0} S\phi(S) = 3.750\ R_o \qquad (3\text{-}15a)$$

In order for the steady-state output attitude to equal the input attitude command of 20 degrees, the value of K_{ϕ_c} (see fig. 3-17) was reduced to 0.267. This is a similar procedure to the development used in the pitch attitude controller. Note that the frequency response (see fig. 3-19a) of Eq (3-15) for $K_{\phi_c} = 1.0$ does not indicate any dominant second-order characteristics which might warrant re-establishing a new final value. Fig. 3-19b exhibits the roll attitude controller frequency response for $K_{\phi_c} = 0.267$. System bandwidth is reduced to approximately 2 rad/sec (0.32 HZ).

The time response corresponding to $\phi_c = 20$ degrees is shown in fig. 3-20. Figures of merit, calculated by TOTAL, are shown in table 3-2.

Having established a linear roll attitude controller and a representative time response suitable for comparison, the next step involves simulation testing. As in the pitch controller simulation, the roll attitude responses to a 20 degree input step command (ϕ_c) were recorded for various K_{ϕ_c}. In order to generate a similar analog controller attitude simulation, K_{ϕ_c} was tuned to exactly 0.267, the same value used in the model. The hybrid simulation response is shown in fig. 3-21. Again, refer to the comparative figures of merit listed in table 3-2. Results indicate very similar characteristics, i.e., T_R within 2%, T_S within 8%, R_V and F_V identical.

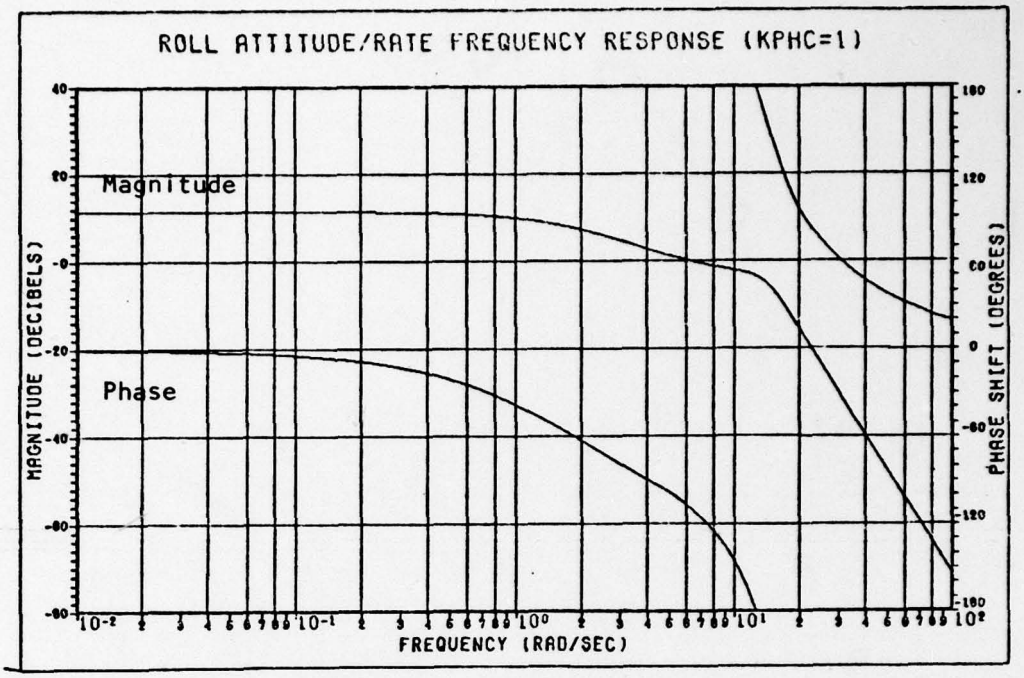


Figure 3-19a: Frequency Response of Roll Attitude Controller, K_{ϕ_c} =1

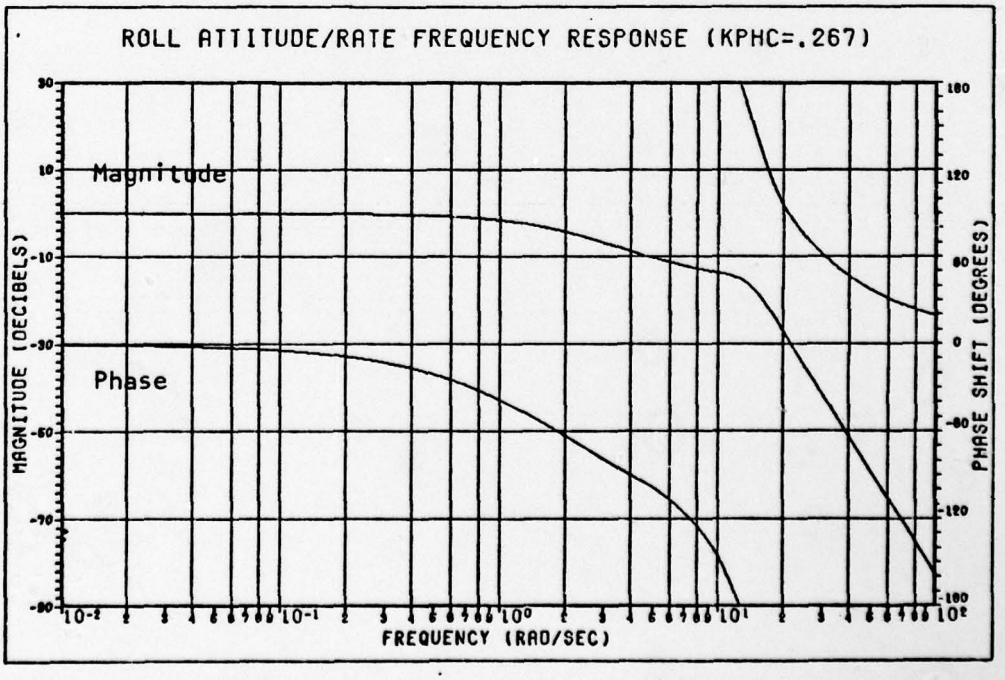


Figure 3-19b: Frequency Response of Roll Attitude Controller, K_{ϕ_c} =.267

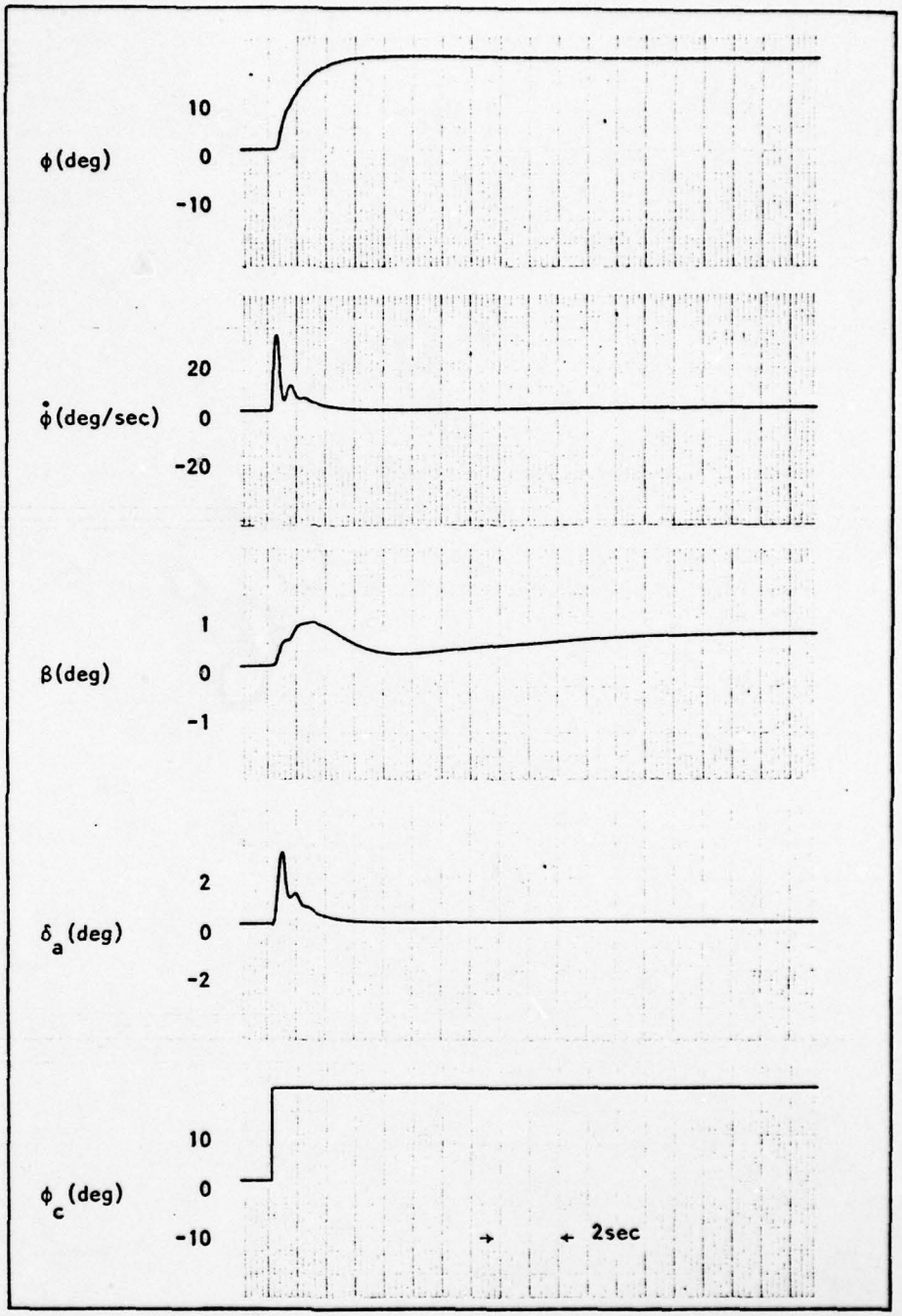


Figure 3-21: RPV Response to 20 Degree Step Roll Right Command(ϕ_c); Roll Attitude/Rate Loops Closed; Analog Controller Engaged

Table 3-2

Figures of Merit: Roll Attitude Controller

$\frac{\phi(S)}{\phi_c(S)}$	T_R	T_D	T_P	T_S	P_V	F_V	
	1.373	Very Large	Very Large	2.600	19.999	19.999	Model
	1.4	Very Large	Very Large	2.4	20.	20.	Simulation (Approx.)

Based upon comparative results, the model appears to be highly correlated with the analog simulation.

Recommendations. The presence of the actuators in the roll attitude controller still degrades the system performance. Specifically, loop gains are restricted by the actuator pole loci to very small values. This inhibits the resultant controller bandwidth, i.e. 0.3 HZ, which, in turn, restricts A/P authority. Actuator replacements and/or feedback compensation, as discussed in the longitudinal section, may be logical recourses to obtain better performance. At present, however, roll attitude controller performance is acceptable.

Linear Yaw Rate Controller (Dutch Roll Damper).

The primary purpose of a Dutch roll damper is to reduce oscillations in roll (ϕ), yaw (ψ) and sideslip (β) angles due to the application of rudder deflections (Ref 17:27). This coupled oscillation is generally referred to as Dutch Roll and as stated earlier, is excited by the rudder (Ref 4). To dampen or otherwise minimize these undesirable effects, yaw rate feedback is employed to deflect the rudder. Fig. 2-5 of chapter II illustrates the analog A/P model with a yaw damper. This controller consists of a Dutch roll damper with lateral acceleration feedback. Through simulation testing, the acceleration feedback, normally utilized for coordination (Ref 4), was found to actually degrade performance. The laboratory agreed and decided to remove it from the analog A/P. As such, this analysis considers only the Dutch roll damper.

The basic Dutch roll controller is depicted in fig. 3-22. Yaw rate is detected by a rate sensor, multiplied by a predetermined gain ($K_{\dot{\psi}}$, KYR), passed through a washout (high pass) filter and then subtracted from the yaw rate command signal ($\dot{\psi}_c$). The generated error signal then drives the actuator as before. The washout filter produces an output only during transient periods. In effect, damping (i.e., feedback) occurs for high frequencies only. The yaw rate signal goes to zero in the steady-state (Ref 4:139).

Following previous methods, the required feedback gain ($K_{\dot{\psi}} = 0.586$, KYR) comes from Appendix B. The RPV transfer function $\dot{\psi}(S)/\delta_r(S)$ (obtained from Appendix A), actuator, and washout filter are combined with $K_{\dot{\psi}_c} = 1$

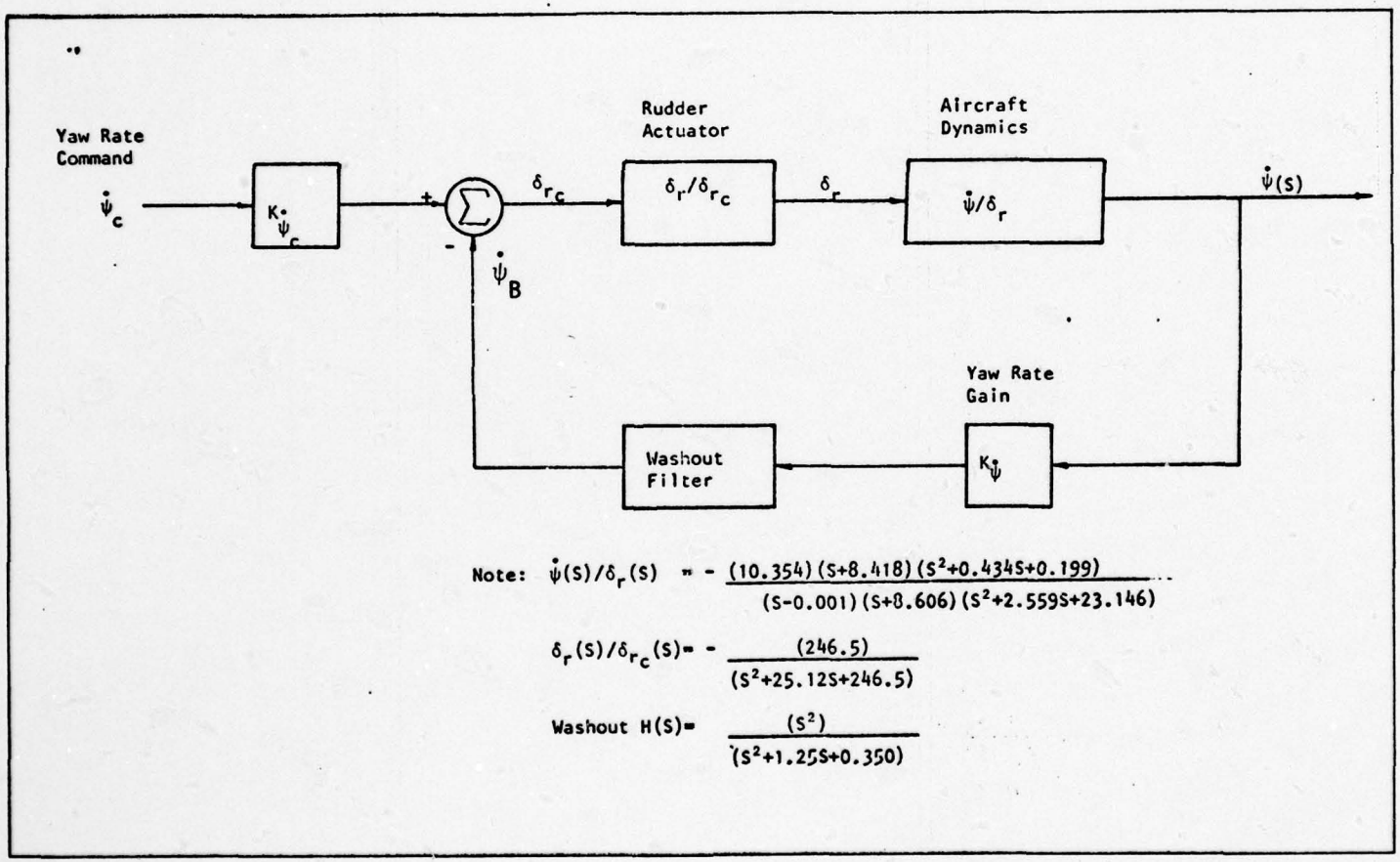


Figure 3-22: Block Diagram of Linear Dutch Roll Damper

to form the open-loop transfer function:

$$\frac{\dot{\psi}_B(S)}{\delta_{r_c}} =$$

$$\frac{(2552)(S^2)(S+8.418)(S^2+0.434S+0.199)}{(S-0.001)(S+0.422)(S+0.829)(S+8.606)(S^2+2.559S+23.146)(S^2+25.12S+246.5)} \qquad (3\text{-}16)$$

which is a type-0 system. The root loci for Eq(3-16) are shown in figs. 3-23a,b.

For the previously specified values of $K_{\dot{\psi}_c}$ and $K_{\dot{\psi}}$, the closed-loop transfer function is found to be

$$\frac{\dot{\psi}(S)}{\dot{\psi}_c(S)} =$$

$$\frac{(2552)(S+0.422)(S+0.829)(S+8.418)(S^2+0.434S+0.199)}{(S-0.001)(S+0.381)(S+8.191)(S+16.55)(S^2+2.978S+2.554)(S^2+9.432S+130.401)} \qquad (3\text{-}17)$$

which reveals that the control system of fig. 3-23 is unstable. Again, the spiral divergence term (S = .001) dictates the "slight" instability behavior with a long time constant (τ_{SD} = 812 sec). For yaw rate stability, however, the higher frequency transients (smaller time constants) are of primary interest. This is especially true since a high-pass filter is involved.

A time response of Eq (3-10) for a selected -3.3 deg/sec yaw rate step command ($\dot{\psi}_c$) is shown in figs. 3-24a,b. Note the unstable affect of the spiral divergence term for t > 2 sec. Also note that the transient

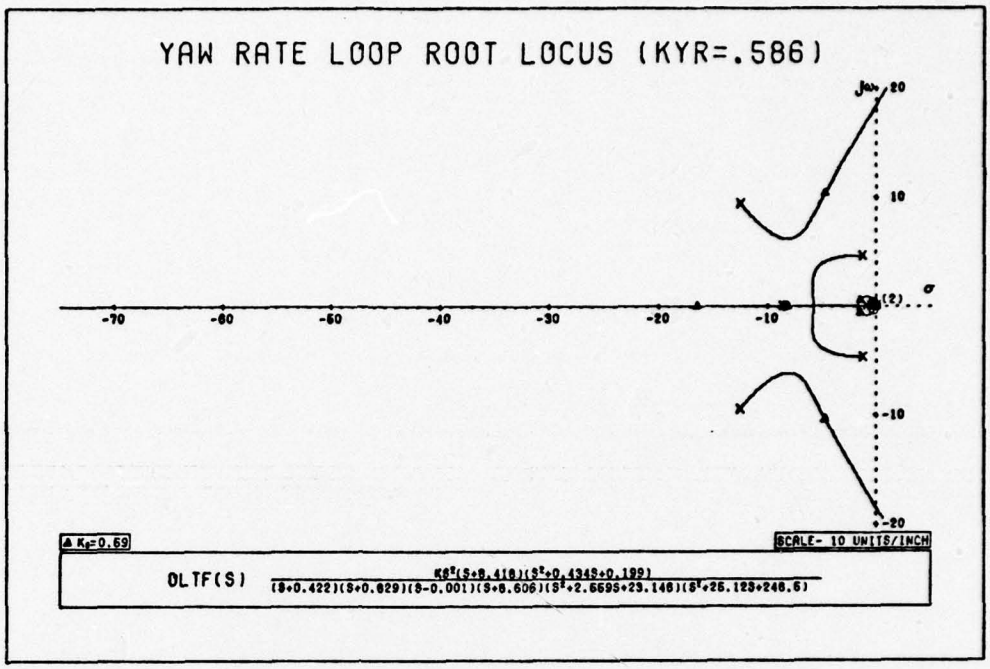


Figure 3-23a: Root Locus of Linear Dutch Roll Damper

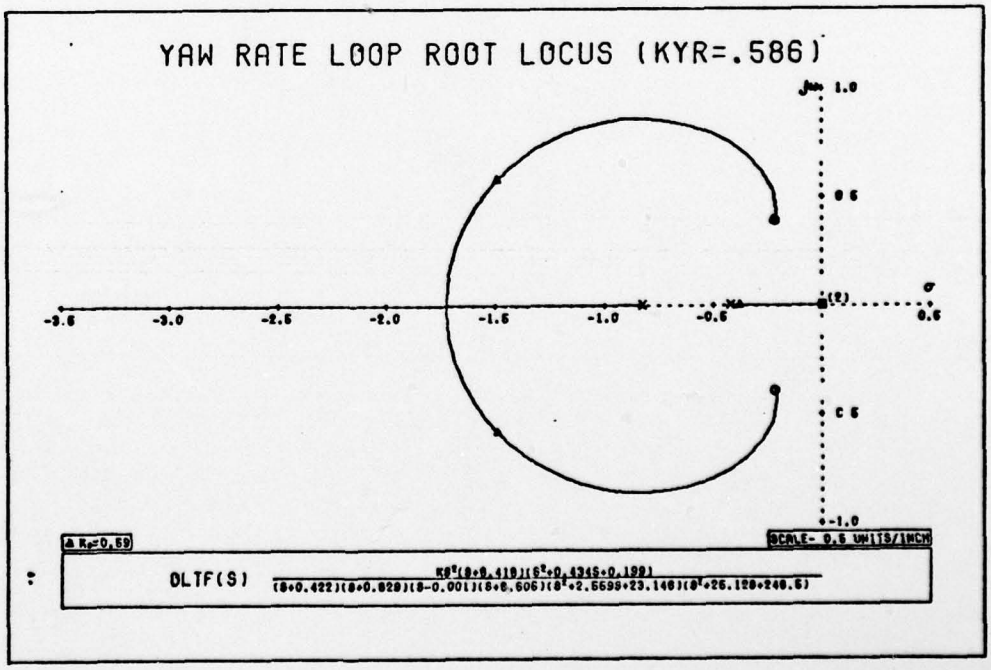


Figure 3-23b: Root Locus of Linear Dutch Roll Damper, Magnified

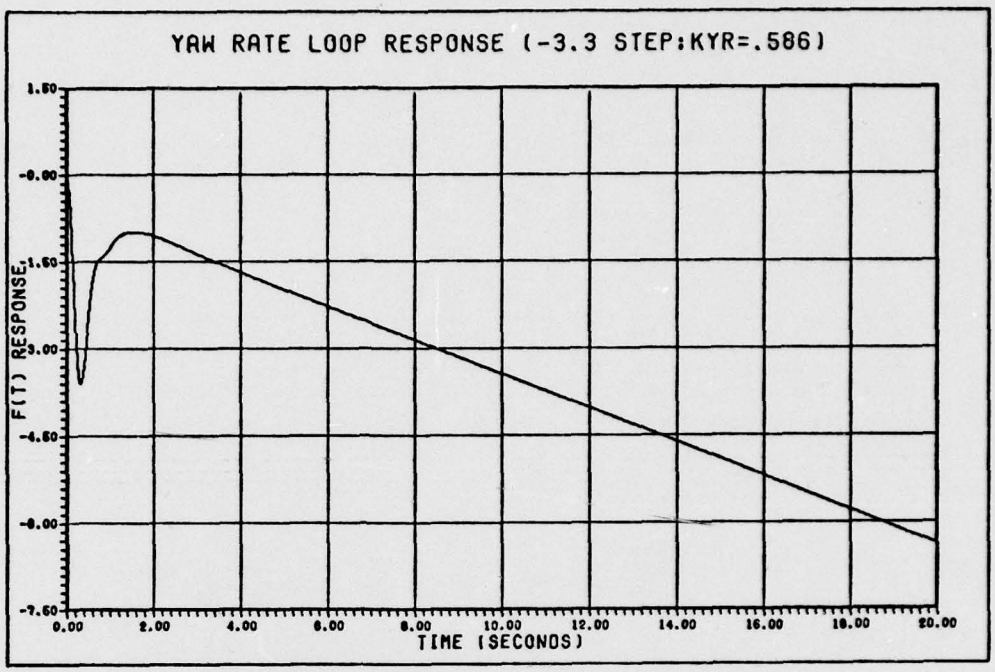


Figure 3-24a: Time Response of Dutch Roll Damper, 20 Sec

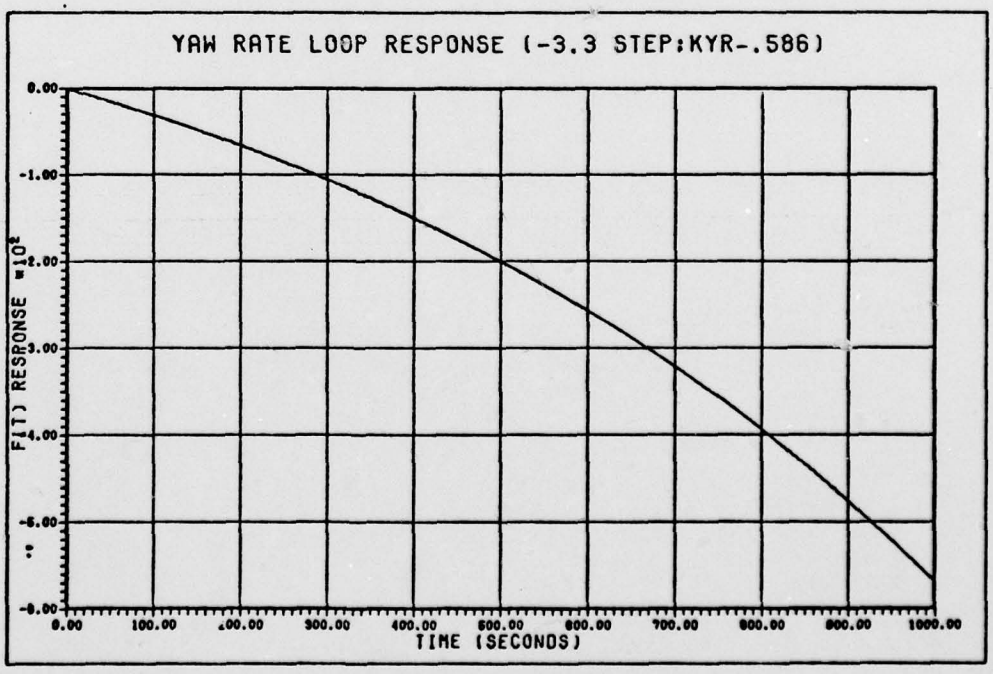


Figure 3-24b: Time Response of Dutch Roll Damper, 1000 Sec

peak value at t = .25 sec approaches -3.3 rad/sec.

A review of the Dutch roll damper frequency response (see fig. 3-25) illustrates the required high-pass filtering with cutoff frequency equal to 1.5 rad/sec (.238 HZ). Also note that the rolloff frequency (ω_c = 16 rad/sec) limits system bandwidth to 14.5 rad/sec (2.31 HZ). Generally speaking, the high-pass filtering cutoff frequency is acceptable. Band rolloff, however, effectively restricts system performance in much the same manner as in the pitch and roll attitude controllers.

Based upon the brief time and frequency response characteristics of the model, analog Dutch roll damper responses were recorded for the desired -3.3 rad/sec yaw rate command (see fig. 3-26). Of particular concern is the output waveform for yaw rate ($\dot{\psi}$). Comparing the model with the analog time responses yields some obvious discrepancies.

---The Dutch roll mode contains the spiral divergence root (S = .001) in the denominator of Eq (3-10). The model time response reveals its subtle effects by the steady-state (ramp) drift effect which occurs after 2 seconds. In order to obtain a good simulation response, the roll axis was coupled to the A/P and acted like a wing-leveller. Little if any rolling occurred due to the rudder input. The gross effect of connecting the roll A/P to the system negates the spiral divergence term from the response. As a result, there is no drift (see fig. 3-26 for $\dot{\psi}$). Even though this discrepancy exists, the controller comparison is not invalidated.

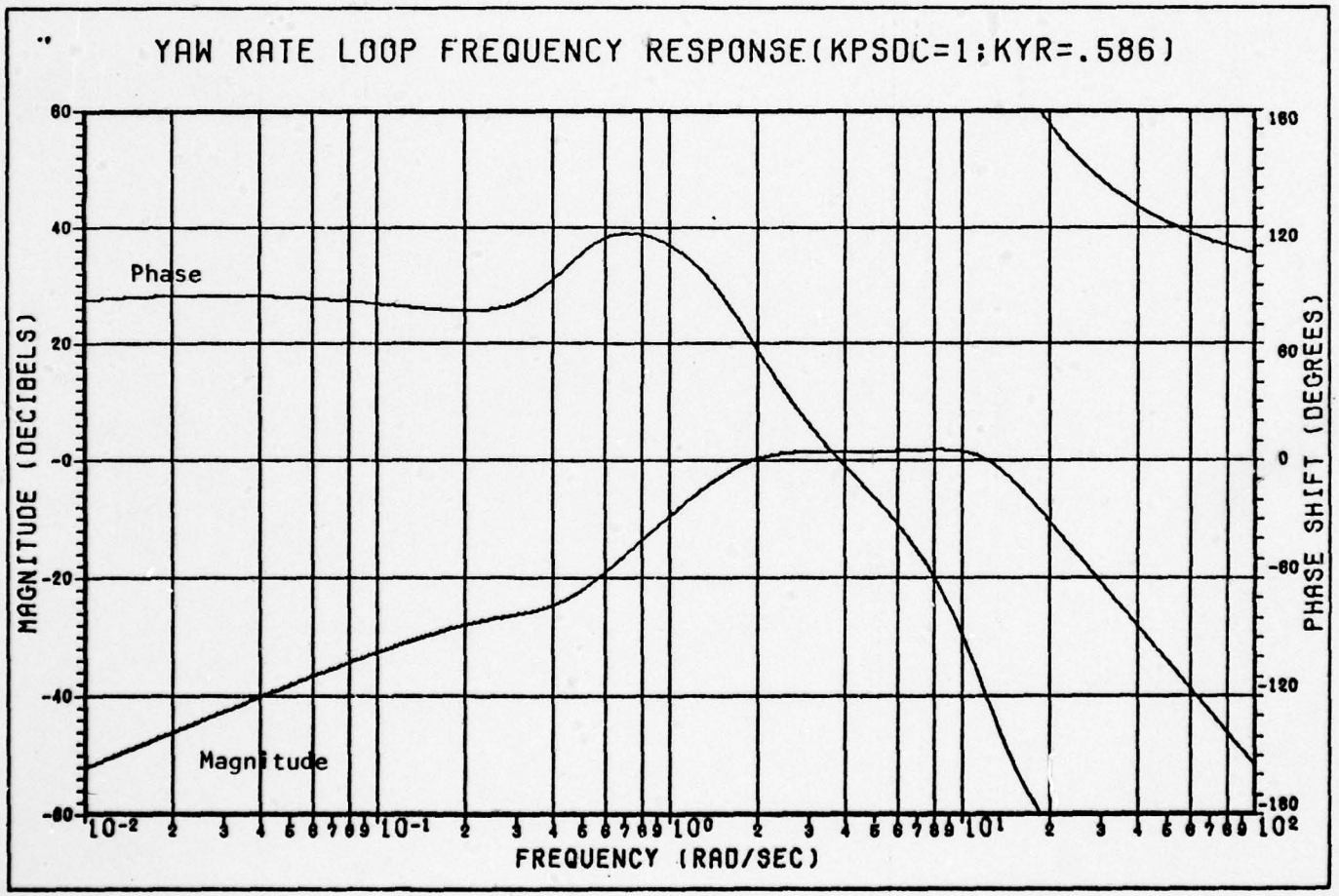


Figure 3-25: Frequency Response of Dutch Roll Damper

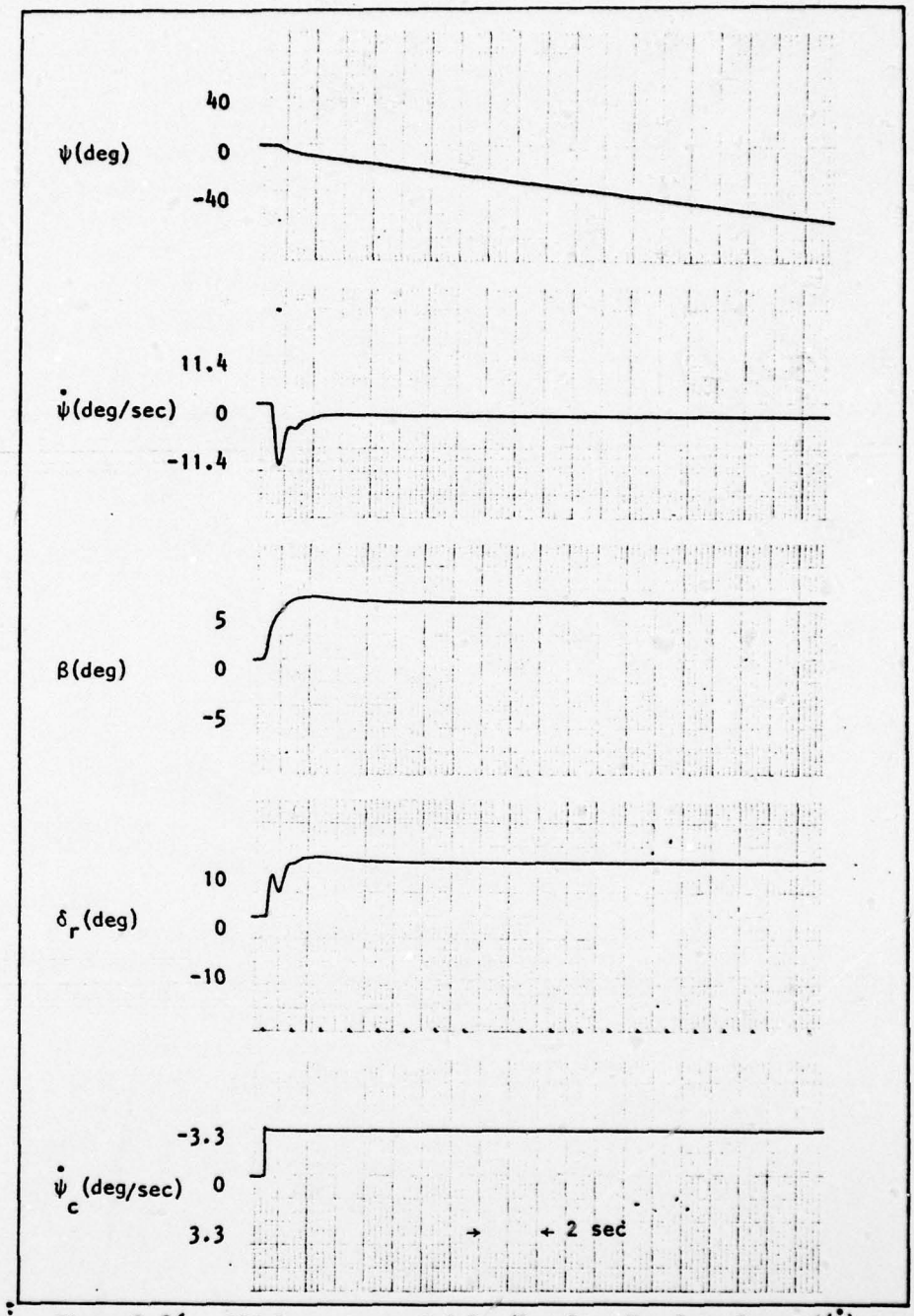


Figure 3-26: RPV Response to 3.3 Deg/Sec Step Yaw Rate Command($\dot{\psi}$); Yaw Rate/Accelerometer Loops Closed; Analog Controller Engaged (Yaw Damper)

---Since yaw rate command gain ($K_{\dot{\psi}_c}$, KPSDC) in both time responses is left equal to unity, one would expect the transients around 0.50 sec to be similar in magnitude. This is not the case, however.

The Dutch roll damper (controller) analysis is somewhat weak because of the problems mentioned above, however, a general sense of confidence, qualitatively speaking, can be exacted. Of all the axes, this one generates the most uncertainty.

Recommendations. Again actuator characteristics inhibit loop gain variations and overall performance. The high-pass filter was chosen for this loop without benefit of control analysis to ascertain its effect on performance. Critical filter time-constant analysis needs to be considered.

A qualitative conventional control analysis has been made of all three axes, comparing the model with the analog controllers. Based upon these results, a fairly good measure of confidence in the analysis has been displayed. The groundwork for further linear control analysis is laid.

IV. Digital Synthesis

Probably the most important theoretical aspect of this study involves the selection of an approach for developing the digital A/P. In the interests of simplicity and realizability, a conventional digitization scheme is chosen so as to produce a digital A/P which performs in a comparable manner to its analog counterpart. As opposed to the more involved "direct digital" design methods which seek to optimize digital system performance, digitization merely converts the continuous (analog) system into a sampled-data system which can perform only as well as the analog system. Digitization, however, is much simpler to analyze and implement (Ref 9). In addition to establishing the basic modus operandi, this chapter discusses analog A/P transformation, significant control system parameters, and digital yaw rate washout filter development. The results of this chapter provide the underlying framework for the actual hardware and software design.

Digitization of an existing analog controller, like that shown in fig. 3-7 of chapter III, yields the pitch axis digital controller exhibited in fig. 4-1. This study assumes that all digital controllers are placed in the feedback paths and incorporate as many loops as possible. Further, only a single sampling frequency is selected for the A/P; multirate sampling is not considered, even though better performance over the single-rate might be achieved. Fig. 4-2, an expansion of fig. 4-1, illustrates that sensor output signals, such as attitude and rate, are selectively sampled by an analog-to-digital (A/D) converter, and then operated upon by the controller to produce a weighted sum. Once the

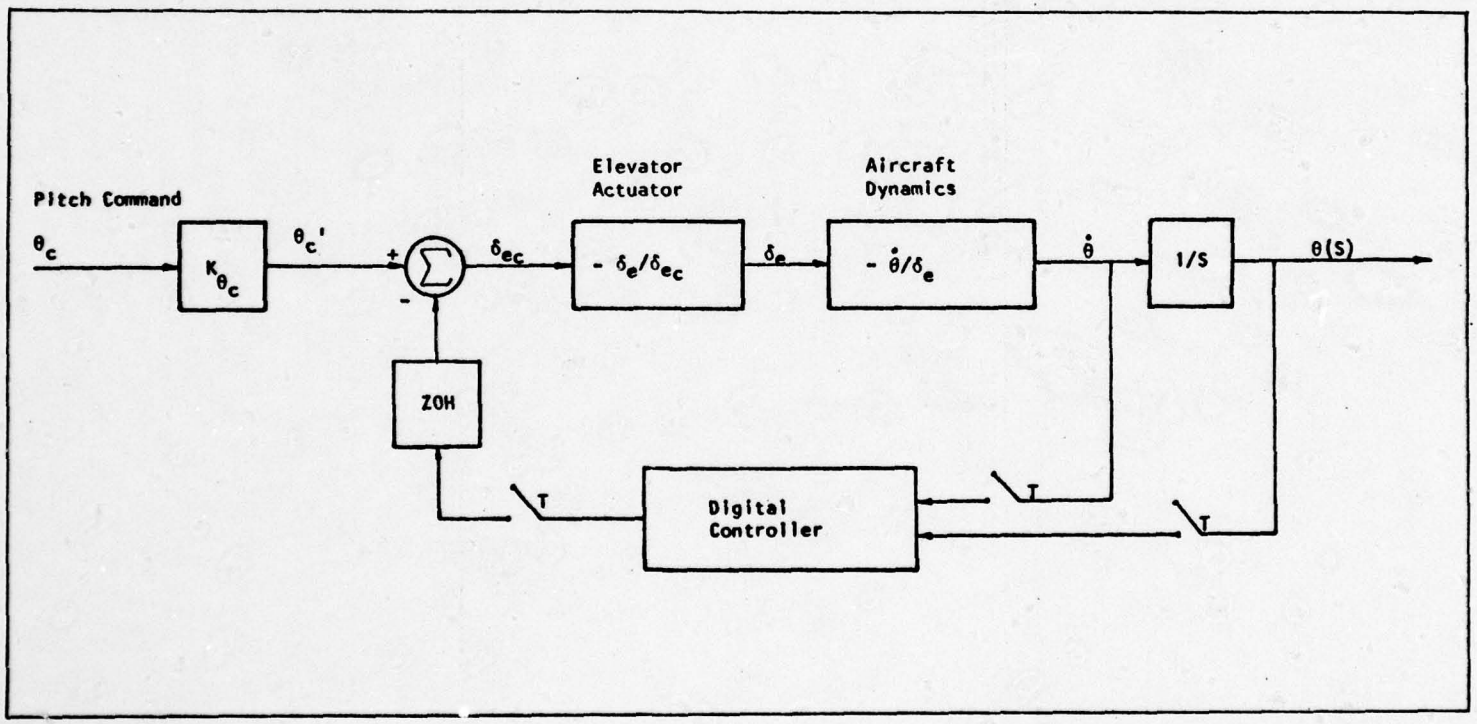


Figure 4-1: Block Diagram of Digitized Pitch Attitude Controller

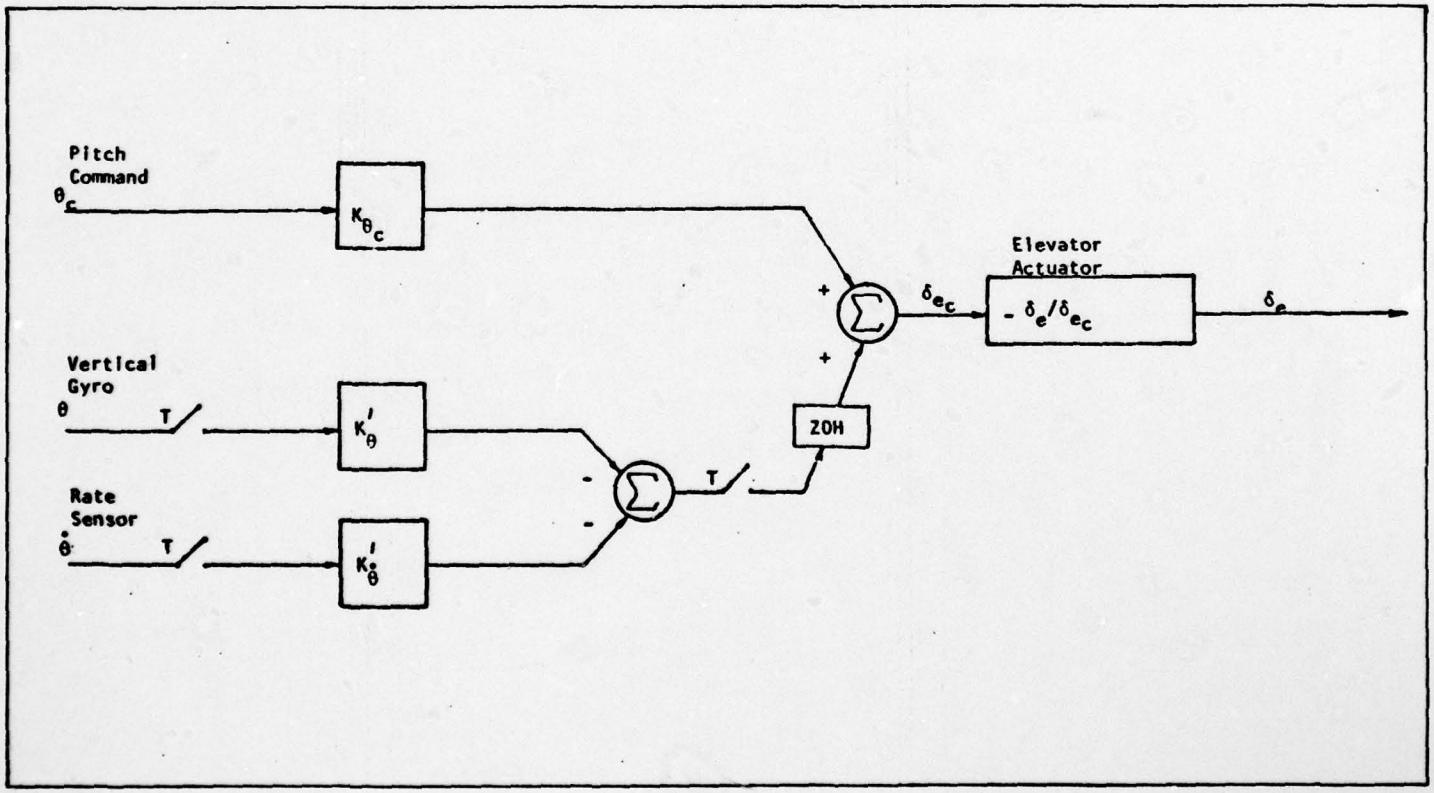


Figure 4-2: Schematic of Digitized Pitch Attitude Controller

processing is complete, the results are restored to analog form via a corresponding digital-to-analog (D/A) converter. The D/A is assumed to contain a sampler and a zero-order hold (ZOH). This is then summed with an analog control input. The resulting error signal commands the actuator to produce a control surface deflection. The digital controller modules shown in figs. 4-1, 4-3 and 4-5 perform the necessary mathematical operations to emulate the analog A/P. For pitch and roll (figs. 4-2, 4-4), essential operations include gain multiplication and signal summing. For yaw (fig. 4-6), a washout or high-pass filter is implemented digitally. As it turns out, filter design becomes the overriding issue which governs such control parameters as word length, sampling time and errors.

The beauty of digital processing is that each control loop can be organized into an overall three-axis A/P under direct control of an operating system. Control of signal acquisition, computations and output follows a definite structure governed by the required sampling frequency.

In order to realize a digital controller, a baseline analysis of the digital control requirements is first established. Significant parameters for this analysis include:

---sampling frequency

---word lengths (A/D, D/A, internal)

---filter design

---other (transport delay, errors, scaling)

A brief discussion of each parameter develops the constraints placed on the digital control system.

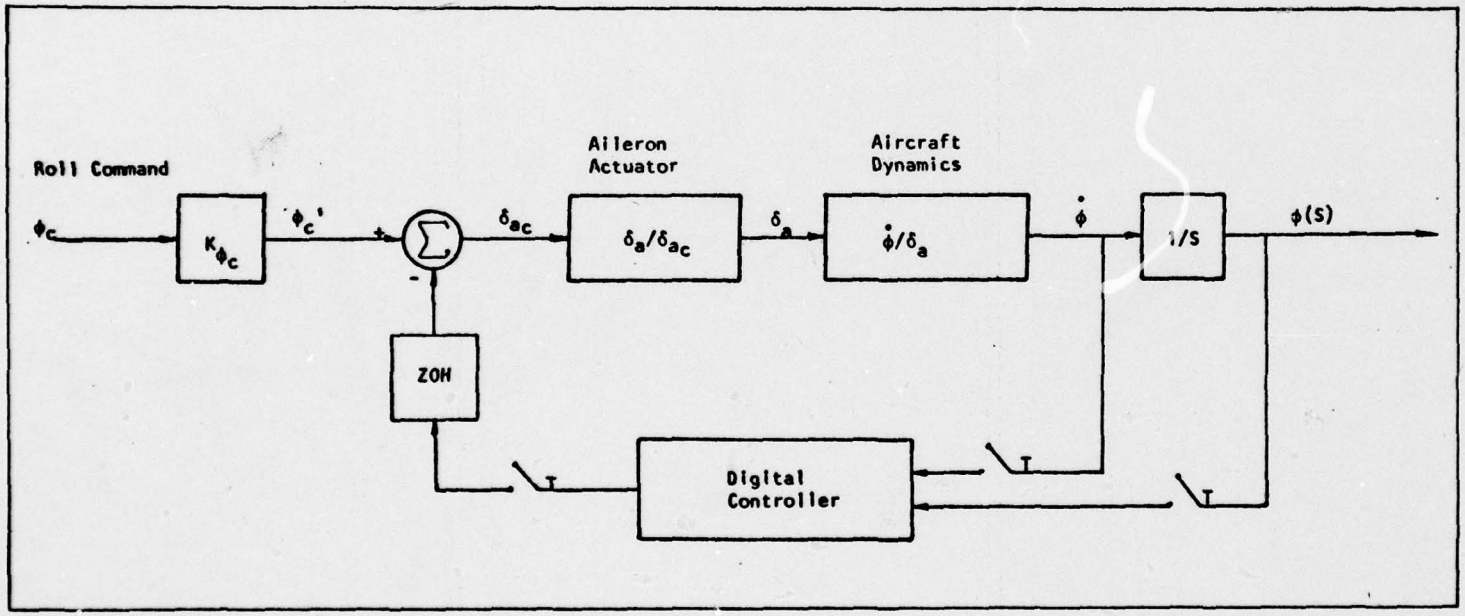


Figure 4-3: Block Diagram of Digitized Roll Attitude Controller

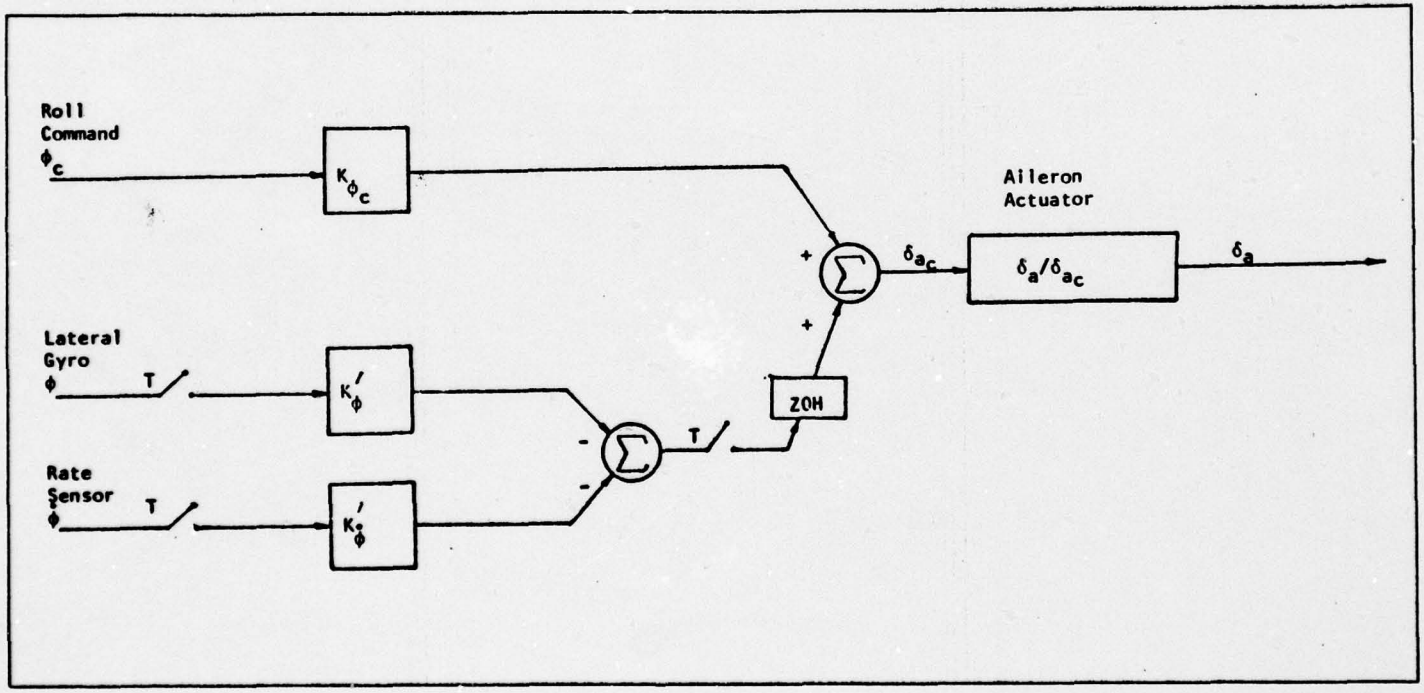


Figure 4-4: Schematic of Digitized Roll Attitude Controller

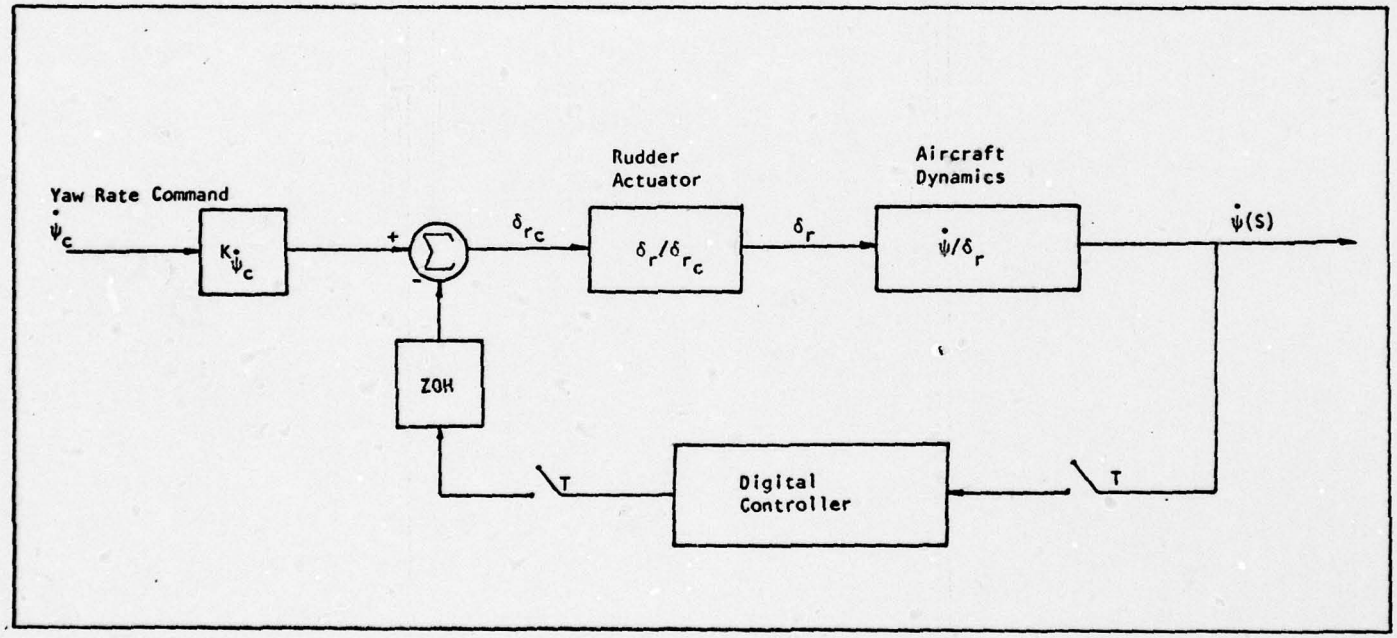


Figure 4-5: Block Diagram of Digitized Yaw Damper

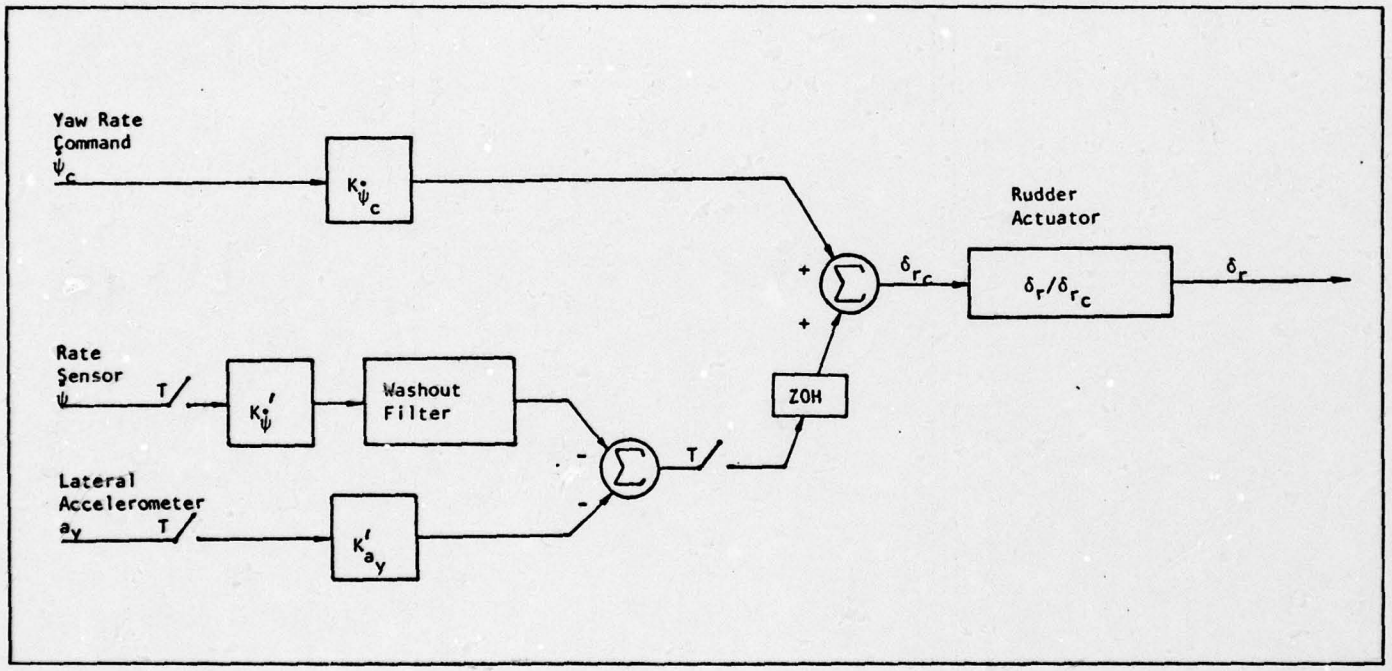


Figure 4-6: Schematic of Digitized Yaw Damper

Sampling Frequency

Since sampling frequency plays such an important role in controller stability, proper determination of such is essential. The Mini-RPV Group indicated verbally that the RPV would not encounter in-flight dynamic frequencies greater than 5-10 HZ. Roskam (Ref42) also substantiates a maximum expected frequency of 10 HZ. Knowing this essential piece of information, then Shannon's (Nyquist's) Sampling Criteria is applied (Ref 27). This theorem states that a signal must be sampled at a frequency (f_s) at least two times the highest frequency (f_m) expected in system (Ref 27).

$$f_s \geq 2f_m \tag{4-1}$$

For practical considerations, one which insures reproducibility, f_s is chosen to be 2-10 times larger than f_m. Because f_m = 10HZ for the RPV, then f_s must be at least 20 HZ. As a first "guess" approximation, f_s was chosen to be 50 HZ. This margin of 30 HZ is not chosen without reason; sampling theory assumes reconstructed data to be represented by infinite wordlength. In the real world, this is not true; finite wordlength dictates performance and restricts signal reconstruction.

Word Length

The three primary areas in a digital system where word length governs cost and performance, include A/D and D/A converters plus internal processing algorithms (Ref 20). Table 4-1 is a summary of sensor and actuator signal output ranges, accuracies, and required A/D resolution. Note that resolution is assumed to be 25% of the accuracy value (Ref 20). Knowing the minimum resolution requirements allows the determination of

TABLE 4-1

A/D Signal Conversion Requirements

Device	Signal Type	Operational Range	Accuracy	A/D Resolution (% of Range) (1)
Vertical Gyro	Pitch Attitude (θ)	+40,-60 deg	±1.0 deg	0.25 deg (.63,.42)
Rate Sensor	Pitch Rate ($\dot{\theta}$)	±40 deg/sec	±0.1 deg/sec	0.025 deg/sec (.06)
Vertical Gyro	Roll Attitude (ϕ)	±45 deg	±1.0 deg	0.25 deg (.56)
Rate Sensor	Roll Rate ($\dot{\phi}$)	±40 deg/sec	±0.1 deg/sec	0.025 deg/sec (.06)
Rate Sensor	Yaw Rate ($\dot{\psi}$)	±40 deg/sec	±0.1 deg/sec	0.025 deg/sec (.06)
Pendulum	Lateral Acceleration (a_y)	±45 deg	±0.2 deg	0.05 deg (.11)
Actuator	δ_c	±45 deg	±0.5 deg	0.125 deg (.28)

(1) Resolution requirement taken as 25% of accuracy requirement (Ref 20)

A/D Wordlength Requirements

Word Length	LSB Resolution	Full Scale Resolution (Volts) (2)
8	.0039063	.019531
9	.0019531	.009766
10	.0009776	.004883
11	.0004883	.002441
12	.0002441	.001221
13	.0001221	.000610
14	.0000610	.000305
15	.0000305	.000153
16	.0000153	.000076
⋮		

(2) Resolution = 2^{-n} where n is number of bits; sign bit is not utilized. Full scale 0-5 VDC.

a minimum acceptable A/D wordlength. The bottom of Table 4-1 clearly shows that eight bits is all that is necessary to faithfully reproduce system signals. Since minimum system cost prevails, an eight-bit A/D converter is desirable, especially since the microprocessor will have an eight-bit wordlength. As will be shown in the next chapter, an A/D with a conversion time (τ_c) of 100 μsec is chosen. KUO (Ref 27) indicates that the A/D conversion time is related to the maximum input frequency (f_{max}) by

$$f_{max} = \frac{1}{(2\pi)(\tau_c)2^{n-1}} \qquad (4\text{-}2)$$

where n = number of bits in A/D.

Substituting the values in Eq (4-2) yields f_{max} = 12.43 HZ. This provides good justification for selection since the A/D will reproduce input frequencies up to 12.43 HZ, well within the RPV bandwidth. If, perchance, higher RPV frequencies are involved, a sample and hold device will be necessary.

The RPV actuator resolution requirements are also listed in Table 4-1. Following a similar development to the A/D case, only eight bits are necessary for its acceptable D/A operation.

Internal wordlength processing requirements are satisfied by sixteen-bit, double precision, fixed-point arithmetic. This is primarily a result of required gain scaling to achieve desired nominal accuracy for the controller. The digital filter, discussed in next section, will be shown to require at least fifteen bits. From an experimental standpoint, it is found that sixteen-bit wordlength performed effectively in the digital control algorithms. This insures a comparable level of performance.

Digital Filter

Generally speaking, a digital filter is a linear shift-invariant discrete-time system that is realized using finite-wordlength (precision) arithmetic (Ref 37:195). This study involves converting the existing analog yaw rate washout (high-pass filter) to a digital filter with similar performance characteristics. Whereas an analog (continuous) filter is represented by a differential equation, a digital filter, as in this case, becomes a linear difference equation. Numerous techniques are cited to perform this transformation (Ref 2;3;6;9;10;20;21;26;27;28;37). No method will provide a set of difference equations which will behave like the continuous analog case in all respects. Ideal transform properties include (Ref 40:A-7):

---stability preservation (assumes infinite wordlength)

---filter cascading capability

---system order retention

---DC gain retention

---impulse response retention

A review of several transform methods was made. Included were first-difference, Tustin, Z-transform, and matched Z-transform. State-variable methods were specifically excluded from consideration based on a trade-off between hardware and software requirements. Based upon the ability to preserve stability, cascading, system order and DC gain, the Tustin transformation was chosen to design the digital filter. It is very popular (Ref 6;48) and easy to use, especially since it exists as an option in TOTAL (Ref 28).

With this in mind, the existing analog high-pass filter,

$$\frac{S^2}{S^2 + 1.25S + 0.35} \tag{4-3}$$

of cutoff frequency equal to 1.40 (ω_c = 1.40 rad/sec), is converted via the bilinear Tustin transform,

$$S = \frac{2}{T}\frac{(Z-1)}{(Z+1)} \tag{4-4}$$

where $T = \frac{1}{f_s}$ = .02 sec, to the Z-domain expression,

$$G(Z) = \frac{(.98762018 - 1.9752404Z^{-1} + .98762018Z^{-2})}{(1 - 1.9751712Z^{-1} + .97530950Z^{-2})} \tag{4-5}$$

Eq (4-5) is then converted to the linear difference equation:

$$Y(k) = .98762018\ X(k) - 1.9752404\ X(k-1) + .98762018\ X(k-2) + 1.9751712\ Y(k) - .97530950\ Y(k-1) \tag{4-6}$$

Based upon an analysis of the analog filter roots, the sampling frequency, and minimum acceptable root error (assumed to be 10%) in the Z-domain transfer function Eq (4-5), a minimum word length of fifteen bits is required to implement all coefficients in the digital filter (Ref 20:64). This discovery, not unexpected, then led to the adoption of sixteen-bit, double-precision arithmetic for the filter algorithm.

Frequency responses for the analog filter (fig. 4-7) compared to digital filters (fig. 4-8, Tustin; fig. 4-9, first-difference) reveal almost no shift of cutoff frequency and no DC gain variation. Again,

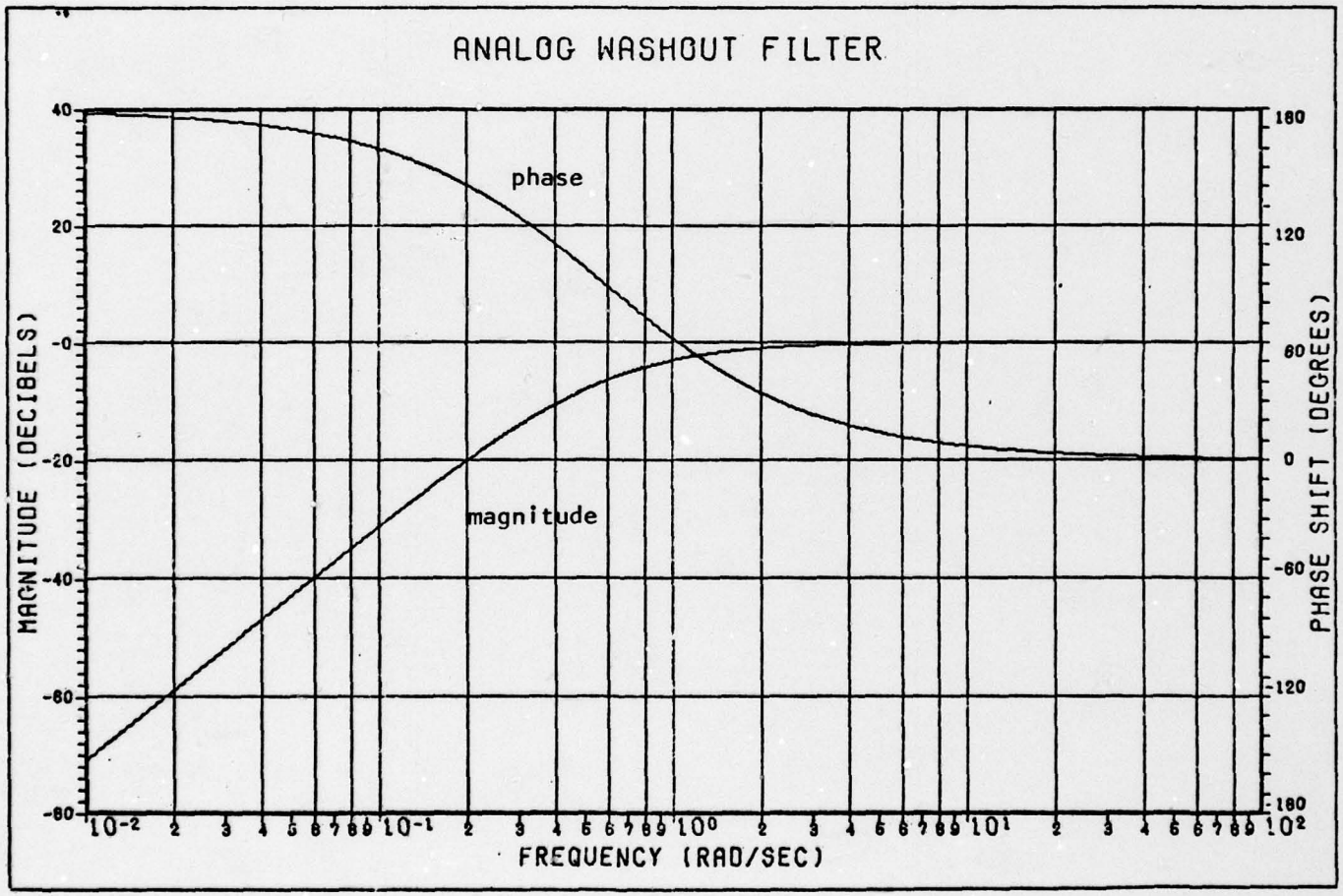


Figure 4-7: Frequency Response of Analog Washout Filter

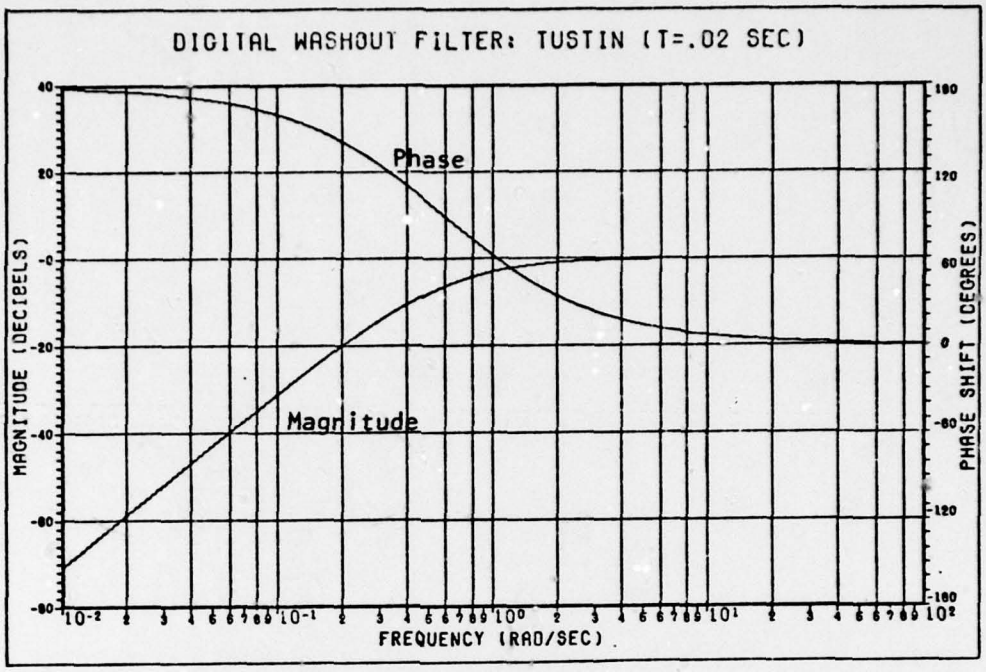


Figure 4-8: Frequency Response of Digital Washout Filter (Tustin, f_s = 50 HZ)

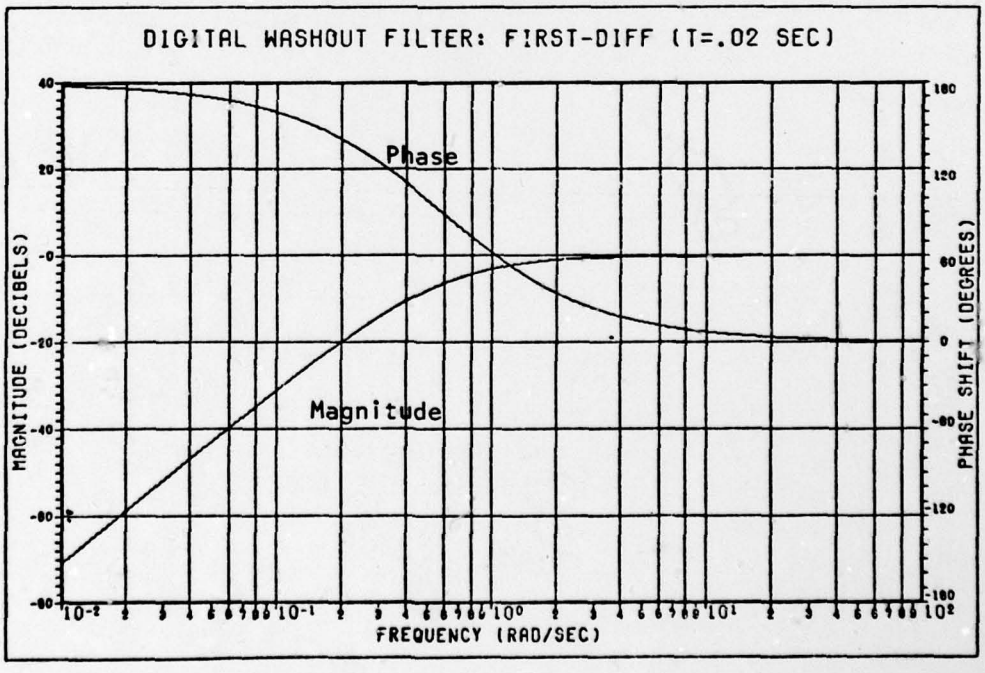


Figure 4-9: Frequency Response of Digital Washout Filter (First-Difference, f_s = 50 HZ)

these responses correspond essentially to infinite wordlength (i.e., 64 bit for Cyber Computer). Because sixteen-bit arithmetic is required, the filter was implemented in software and a frequency response generated to determine performance effects. The experimental results are discussed in chapter VII. Note, the Tustin transformation may require that the analog filter cutoff frequency be "prewarped" so as to be accurately reproduced in the discrete domain (Ref 37). This facet of filter development is not significant in this application. The digital filter was only designed for f_s = 50 HZ. If the sampling frequency needs to be changed, so must the filter constants. Chapter VI will describe in detail the software implementation of the recursive high-pass filter.

Other

Although not specifically addressed in this study, additional factors must be considered in a digital control analysis. These include (Ref 20):

---studying the effects of transport delays due to system components (A/D, D/A, etc.) on performance

---investigating the effects of bit truncation and rounding associated with wordlength (Chapter VI briefly discusses the topic)

---describing the effects of a noisy environment on system performance; this may dictate that signal "conditioning" be a factor

---performing sensitivity analysis to ascertain best or desired performance (consider wordlength and sampling frequency)

This chapter establishes the basic control theory philosophy and parameters needed to continue with the digital A/P mechanization in hardware and software.

V. Digital Mechanization

Fig. 5-1 is a diagram of the layout of the prototyping board used to construct the digital autopilot circuit for interfacing with a hybrid simulator. This chapter will describe how the various components were selected and how the associated circuits operate.

Hardware Selection

Microprocessor. Although hardware selection involves many components, the critical issue is the selection of the microprocessor. The following items were considered in examining the various microprocessors:

1. Basic CPU Data Word Length. The basic selection process is essentially the choice between eight-bit and sixteen-bit data word length. At the time this project was started, there were only a few sixteen-bit processors available, with others due to be on the market soon. Since one of the primary objectives of this project is a very low-cost system, the sixteen-bit processors were not considered since their current cost (plus support items) is considerably higher than eight-bit processors. It should be noted that this condition is not likely to last forever, and a re-evaluation of the feasibility of using a sixteen-bit processor should be made in the near future due to their increased performance capability. An eight-bit processor can still meet the flight control algorithm word-length requirements by means of multiple-precision arithmetic. The disadvantage of this method is reduced execution speed, but if excess capability exists, few

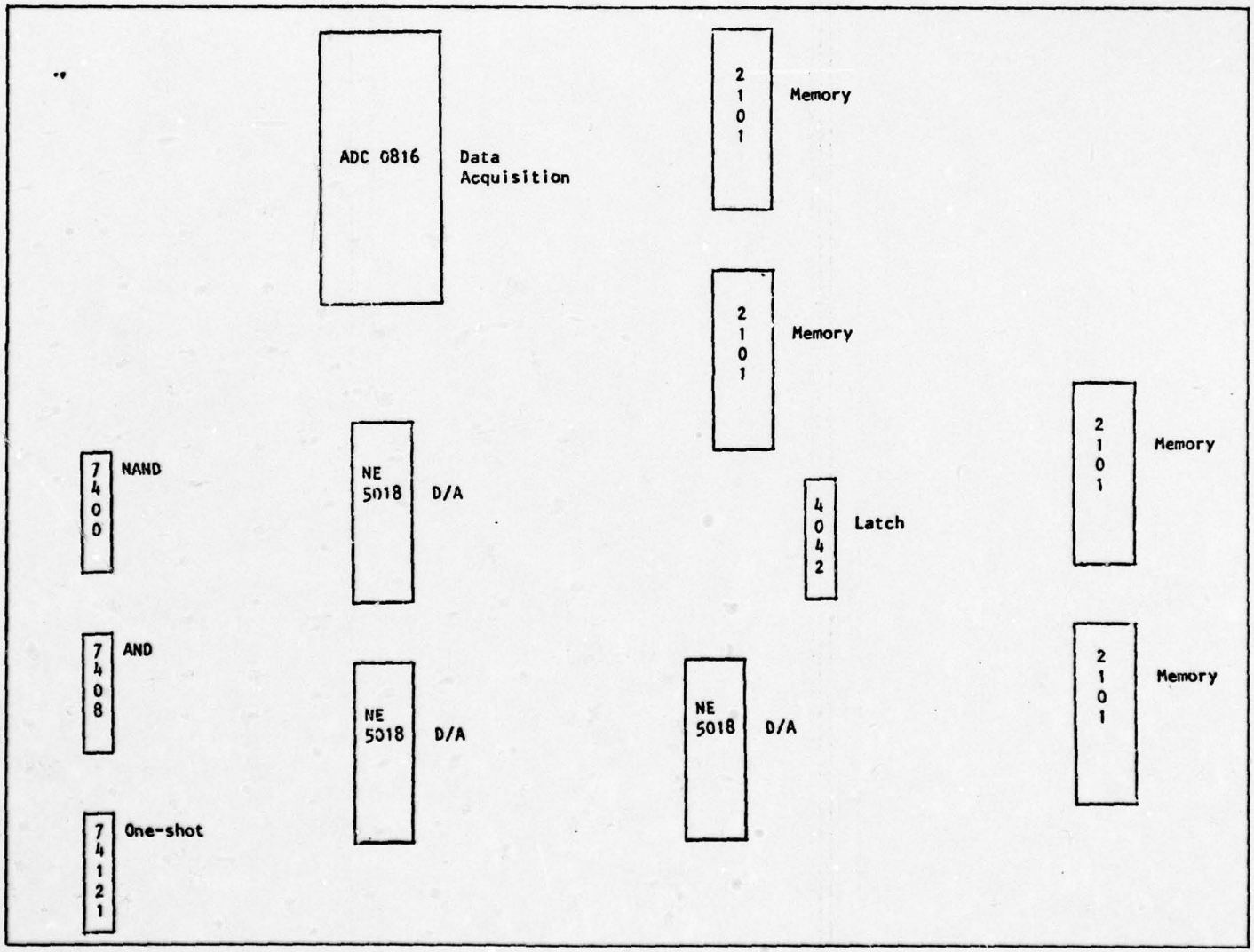


Figure 5-1: Prototyping Board Layout

problems will be encountered.

2. Power Consumption. RPV electrical power is derived from an engine alternator/nickel-cadmium battery combination. Although no strict upper limit of autopilot power consumption has been established, the lowest possible power consumption is desirable since other systems (telemetry, tracker, etc.) are constantly being added that compete for the limited amount of power available.

Since the RPV power system is a +28V supply, it is therefore desirable that the microprocessor not require any negative voltages. Although negative voltages can be generated with additional circuitry, the constraints of cost, size, and weight dictate that a single voltage processor be used.

3. Speed. The autopilot algorithms contain many multiplications that are very time-consuming when accomplished in software. Although a dedicated high-speed hardware multiplier could be used, the cost, size, weight, and power constraints make this option undesirable. Thus, short instruction times (i.e. high speed) is an important factor in this application.

4. Cost. Most eight-bit microprocessors are in the same general price range. However, the support circuitry required to operate them may vary widely. Thus, the complete system required to implement an autopilot function should be examined to insure lowest cost commensurate with desired performance characteristics. Since hardware cost is largely a function of chip count, the same argument can also be used to minimize size and weight.

A good summary of the characteristics of the various microprocessors is found in Ref. 7, page 56. The approach taken in choosing a processor was to define a set of desirable characteristics, select all processors that met these characteristics, then further evaluate the selections to arrive at a final decision.

The initial characteristics desired were: eight-bit word length, single chip CPU, single power supply, on-chip clock, and interrupt and DMA capability. Three processors met the above characteristics: Intel 8085, Motorola M6802, and RCA CDP1802. In terms of speed, the 8085 is fastest (1.3 μsec shortest instruction), the 6802 is second (2 μsec) and the 1802 third (2.5 μsec). The 1802 has a decided advantage in terms of power consumption, 1.6 mA versus 100 mA for the 6802 and 170 mA for the 8085. Being a CMOS structure, the 1802 also has an advantage in terms of high noise immunity. This could be particularly important in light of the severe environment in which the autopilot must work (near an ignition system, alternator, servo motors, downlink telemetry transmitter, and a microwave tracker).

It appears that none of the microprocessors has an overwhelming advantage over the others; each has its own strengths and weaknesses. The RCA CDP1802 was finally chosen based on several factors: low power consumption, good noise immunity, system build-up with a minimum of support items (i.e. low chip count), easily implemented input/output programming, indirect oriented architecture that makes programming without an assembler reasonably easy, previous programming experience with the 1802, and ready availability of a microcomputer board based on the 1802.

Analog to Digital Converter. While the characteristics of available analog to digital (A/D) converters were being reviewed, National Semiconductor introduced the ADC 0816, a complete data acquisition system on one chip (Ref 1). Its features include: eight-bit successive approximation conversion, maximum of 100 μsec conversion time, sixteen-channel analog multiplexor with latched address decoders, an output latch and Tri-State output buffer, and a single 5V power supply requirement that consumes only 15 milliwatts. This is the first time that such a circuit has been available in monolithic form (and at low cost), and as such, was a natural choice for this project. Another plus (and somewhat unique) feature of the ADC 0816 is the use of 256R resistor-ladder netword to insure a monotonic response, an important feature in a control application since non-monotonic response could cause unwanted positive feedback and potential oscillations (Ref 8). As described in Chapter III, an eight-bit A/D is sufficient for system requirements.

Digital to Analog Converter. A similar situation to the one encountered while searching for an A/D converter was encountered during the search for a suitable digital to analog (D/A) converter. Signetics introduced a D/A converter system on a single chip, the NE 5018 (Ref 14). It contains an eight-bit D/A converter, an input latch, a voltage reference, and an output amplifier. The main drawback of the NE 5018 for this application is that it requires both positive and negative supply voltages, a situation which is undesirable since the aircraft has only positive voltages available and required negative voltages will have

to be generated. It was decided that the cost, space, and weight savings of using the NE 5018 would more than offset the additional circuitry required to generate a negative voltage. This is particularly true since the savings of using the NE 5018 are multiplied by three since three D/A's are required. It is anticipated that an equivalent part requiring only a positive voltage will eventually be introduced by some manufacturer, and the switch to that part should cause a minimum number of changes.

Hardware Multiplier. As stated earlier, software multiplications are very time-consuming. Initially, there was some concern that the microprocessor might not operate fast enough to accomplish the control algorithm using software multiplication, so a hardware multiply chip was obtained for use if this was the case. (The multiply chip was not needed.) The multiply chip selected was the TRW TDC 1008J (Ref 36). This is a high performance monolithic chip with a speed much higher than this application requires (70 μsec), but is the best choice at the present time. If a multiply chip is needed in the future due to additional functions incorporated into the autopilot, a soon-to-be-released chip, the RCA CDP1855 multiply/divide unit, should be ideal for this application due to lower cost, power consumption, and compatability with RCA 1802/1804 MPU architecture.

Hardware Development

Microcomputer. An "ELF II" microcomputer kit was used as the microprocessor-based system for prototyping (Ref 19). This board

contains a hex keyboard, two-digit hex LED readout, 256 bytes of RAM, 5 volt power supply, and a region of PC board capable of accepting five expansion board sockets. One such board, called a "Giant Board", was utilized (Ref 23). Features of the "Giant Board" that are useful in this application include: a 256 Byte ROM operating system that allows direct examination of any memory location, direct change of any memory location, program execution start at any memory location, a cassette read and write (software UART function), and an input and output address decoder. Both the "ELF II" and "Giant Board" (schematics are included in Appendix C) contain additional capabilities that were not used in this application.

The "ELF II" uses a 3.58 MHZ crystal and a divide by two circuit to obtain a 1.79 MHZ clock frequency. This is used to obtain compatability with a TV interface IC on the board. Since the TV interface is not used in this application, the crystal was changed to 5.16 MHZ to obtain a 2.58 MHZ clock frequency. The clock frequency may be increased to a maximum of 3.2 MHZ when the CDP 1802 is operated at 5 volts (which it is in this design).

An Elite board (prototyping board) was used to mount the remaining components required to construct the autopilot. Cables were used to connect the "ELF II" and Elite board. A prototyping board strip was glued to the "ELF II" in order to make connections to it.

A/D Circuitry. Fig. 5-2 shows the circuitry used to interface the ADC 0816 data acquisition chip to the system. The multiplexor address is routed over the four LSB's of the data bus. Therefore, to send an address to the multiplexor a value of 00 to 0F is sent to

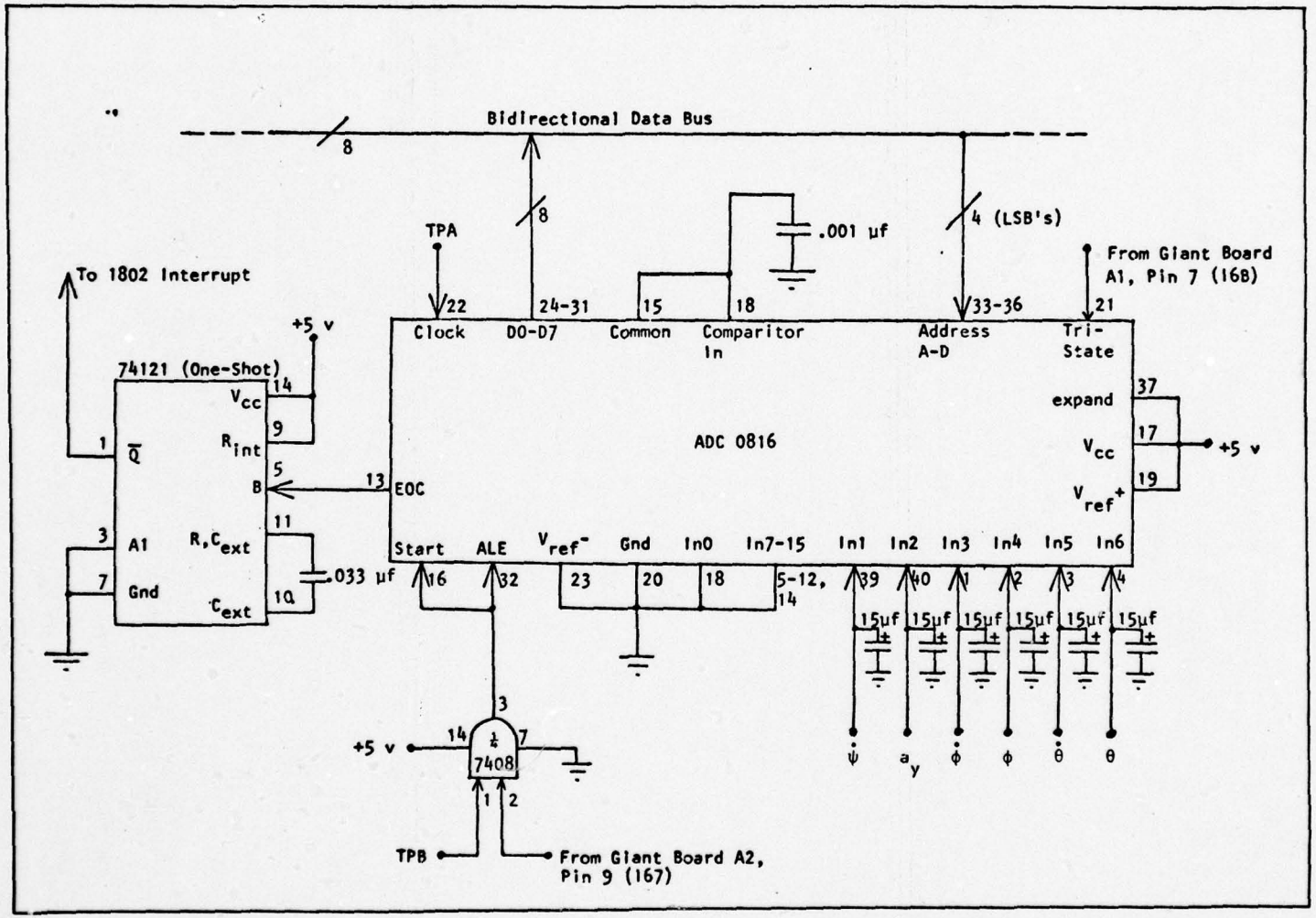


Figure 5-2: Analog to Digital Converter Circuitry

provide an address of input 0 through input 15, respectively. The instruction used to output the address is 67 (value of 7 on I/O address lines) which also latches the multiplexor address and starts the conversion process. This is accomplished by using the I/O address decoders on the "Giant Board" as an indicator of instruction 67. The output of the "Giant Board" I/O address decoder is ANDed with TPB (a timing signal from the CPU) to insure the multiplexor address is stable when the multiplexor latches the address.

A one-shot, SN 74121 (Ref 50), is required between the A/D's end of conversion (EOC) signal and the CDP 1802's interrupt input. If the two were connected by just an inverter, the interrupt signal would still be present when the interrupt routine completed, and the program would be locked in an interrupt routine loop. The output of the one-shot must be long enough to insure the executing instruction completes (> longest instruction time), but shorter than the interrupt routine. The one-shot pulse width is given by (Ref 50):

$$t_w \text{ (out)} = .7\ C_{ext}\ R_T \qquad (5\text{-}1)$$

Using C = .033 μF and the internal 2K resistor results in a pulse width of approximately 46 μsec, during which approximately 7.5 instructions will execute (with a 2.58 MHZ clock frequency). Since the interrupt routine to input data from the A/D is 10 instructions in length, the one-shot will have removed the interrupt condition before the interrupt routine completes. Note that if the 1802 clock frequency is increased, the value of the capacitor used with the one-shot may have to be decreased so that the interrupt routine does not complete before the one-shot times out.

A .001 μF capacitor was connected between the multiplexor out (pin 15)/comparator in (pin 18) loop and ground. This was necessary to filter out noise being generated in the multiplexor (switching noise). Without this capacitor the A/D produced a one-bit jitter in its output. Also, a 15 μF capacitor was connected between each analog input and ground to filter out any noise that might be generated by the sensors or picked up in transmission. As determined experimentally, the corner frequency of this filter is about 150 Hz.

D/A Circuitry. Fig. 5-3 shows the circuitry used to interface the NE 5018 D/A converter circuitry with the system. The basis of the circuit is the recommended circuit given in the Signetics application bulletin (Ref 14). An external feedback resistor of 4.7 K was added to the output amplifier (pins 18 and 20) to reduce the output voltage range to about 5 volts. The adjustment to obtain exactly 5 volts full scale is made at the potentiometer connected between pins 13/14 and ground.

The Latch Enable ($\overline{LE}$) signal is obtained from the appropriate output of the I/O address decoder on the "Giant Board". Instruction 63 latches a value in the yaw axis D/A, instruction 65 into the roll axis, and instruction 66 into the pitch axis. The signal from the I/O address decoder is ANDed with TPB (timing signal from the CPU) to insure the input data from the bus is stable before the data is latched. The result is inverted (thus a NAND is used) to obtain the correct polarity.

A 1K resistor is connected between the D/A output and ground to insure a constant load on the output amplifier. Also, a 10 μF capacitor is connected between the output and ground to filter any noise,

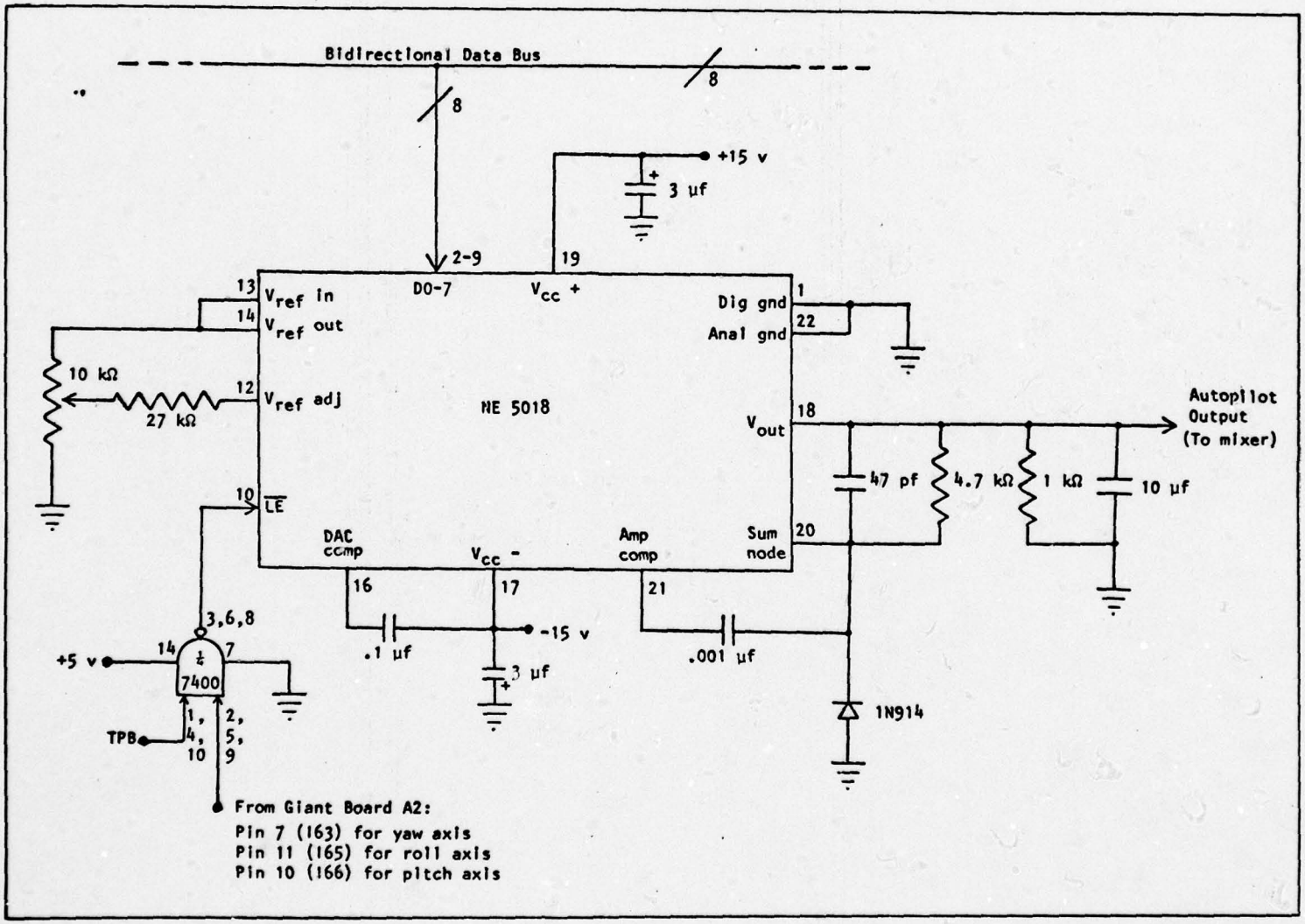


Figure 5-3: Digital to Analog Converter Circuitry

such as major transition glitches (Ref 13).

Memory. The software requires 656 eight-word bytes of memory. Since the basic "ELF II" system contains only 256 bytes, two additional blocks of 256 bytes each had to be added. Fig. 5-4 shows the circuitry used. Since the CDP 1802 has only eight address lines, time multiplexing must be used to send a sixteen-bit address. The address upper byte is sent out and latched, then the address lower byte is sent out. With full decoding of the upper byte, up to 65536 bytes of memory can be addressed. However, full decoding of the address upper byte can be avoided if eight or less blocks (1 block = 256 bytes) are all that is required. The bits of the upper byte can be directly latched if only one bit is high at a time. With this scheme, additional blocks can be addressed as 01XX, 02XX, 04XX, 08XX, 10XX, 20XX, 40XX, and 80XX. In this application, Bits 0 and 2 were latched in a CD4042 D flip-flop (Ref 12) so that the two additional blocks are addressed as 0100-01FF and 0400-04FF. The original 256 bytes on the "ELF II" board are addressed as 0000-00FF and have no address upper byte selection (selected all the time). Thus, this block must be disabled whenever either of additional two blocks is selected. This is accomplished by connecting the inverted output of each latch, $\overline{Q_n}$, to the chip select line of the "ELF II" memory via a diode. The "ELF II" chip select pin is normally held high by connecting it to +5 volts through a 10K resistor (on the "Giant Board"). Whenever either of the additional blocks is not selected, $\overline{Q_n}$ is high and has no effect on the "ELF II" memory. However, when 01XX or 04XX

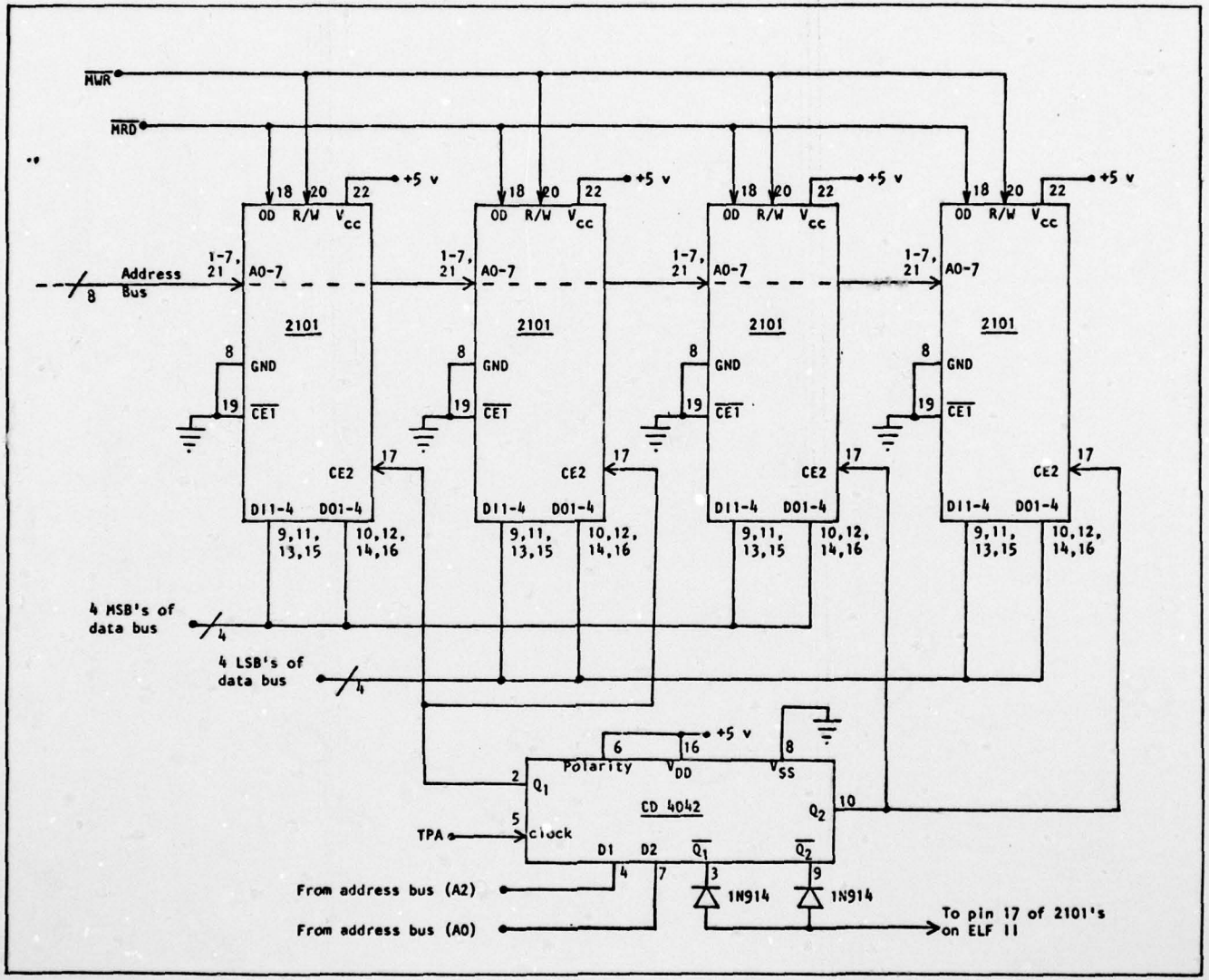


Figure 5-4: Memory Expansion Circuitry

is selected, $\overline{Q_1}$ or $\overline{Q_2}$ go low and ground the chip select pin of the "ELF II" memory, thus disabling it. The diodes act to provide an OR function (diode logic) between the selection of additional blocks of memory.

Timer. Fig. 5-5 is the circuit used for precise control of the sampling rate (T). It is a standard NE 555 circuit from the Signetics applications note for an astable (free-running) pulse generator (Ref 49). After completing an iteration through all three axes, the program tests $\overline{EF3}$ (which is connected to the timer circuit) and waits for the output to go low. When a low is detected, the program branches back to the beginning to repeat the sequence. The time the output is low must be long enough to insure detection (> 2 instruction times), but shorter than the time to execute the complete code of one pass through the program. The output low time, t_2, is given by (Ref 49):

$$t_2 = .693\ (R_B)\ C \qquad (5\text{-}2)$$

Using C = .22 μF and R_B = 2.7K, then:

$$t_2 = 411\ \mu sec$$

which is long enough for approximately 66 instructions to execute. The total time, $T = t_1 + t_2$, should = 20 msec for a 50 HZ sampling rate. The output high time, t_1, is given by (Ref 49):

$$t_1 = .693\ (R_A + R_B)\ C \qquad (5\text{-}3)$$

Since $t_1 = T - t_2$ = 20 msec - 411 μsec = 19.59 msec, then

$$R_A = \frac{t_1}{.693C} - R_B = 126\ K$$

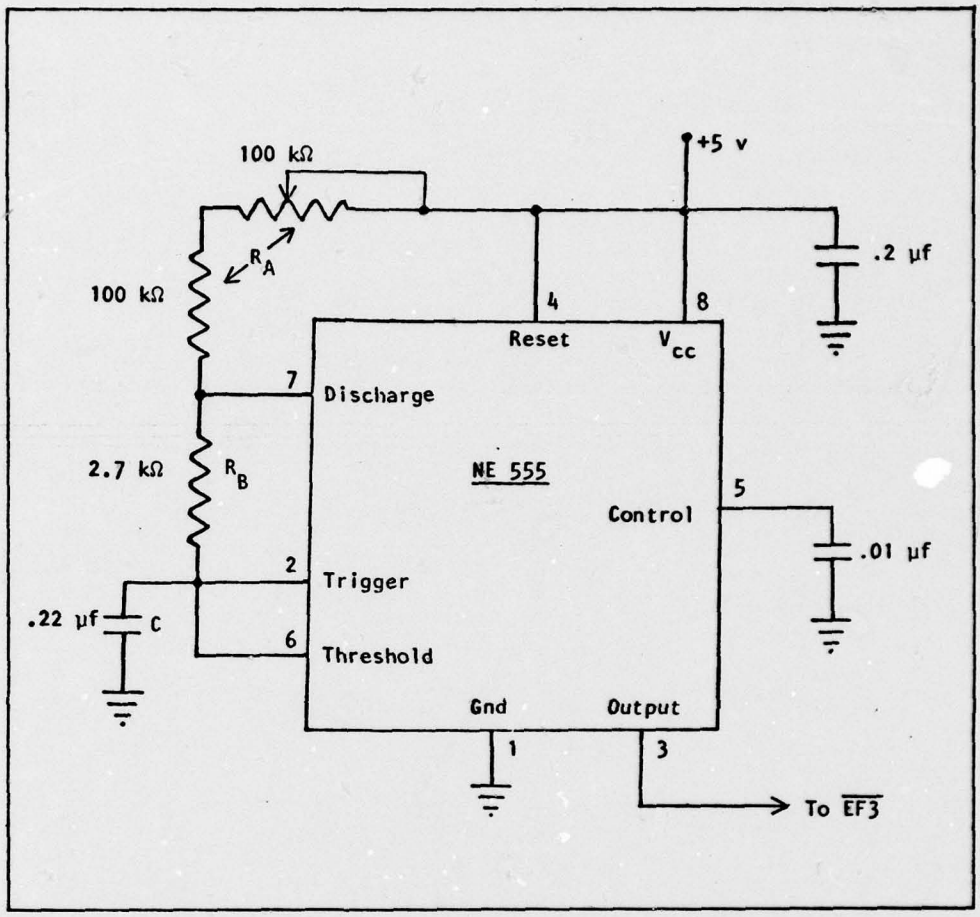


Figure 5-5: Sampling Rate Timer Circuitry

R_A is composed of a 100K resistor and a 100K potientiometer for precise adjustment. Note: Jumper J8 on the "Giant Board" must be removed in order to utilize $\overline{EF\text{-}3}$.

Power. For simulation purposes, a standard tri-voltage lab power supply was used to obtain +5, +15, and -15 volts. For actual proto-type construction, it is recommended that three-terminal regulators be used to obtain +5 and +15 volts, and a Burr-Brown Model 700 isolated DC/DC converter be used to obtain -15 volts from +15 volts (Ref 16).

Support Circuitry. For simulation purposes, TTL parts were used for the various random logic gates required (Ref 50). For prototype construction, it is recommended they be replaced with CMOS equivalents for lower power consumption and better noise immunity (Ref 12).

Along with the software developed in the next chapter, the circuitry described in this chapter was used to test the feasibility of a microprocessor-based RPV autopilot. A hybrid simulation of the RPV dynamics was used and test procedures and results are described in Chapter 7.

VI. Software Development

Transformation of the general autopilot algorithms into computer software is conceptually a simple process, but in practice can present some significant problems due to word length, CPU speed, etc. The purpose of this chapter is to describe such a transformation. The first item to be described is the derivation of a single axis autopilot flow chart. Then the flow chart for the washout (high pass) filter in the yaw axis is developed. Finally, the complete 3-axis (including yaw filter) flow chart is developed by combining the washout filter and three similar single axis routines into a single program. The CDP 1802 program listing is in Appendix E.

Autopilot Loop Algorithm Development

A basic autopilot block diagram is shown in fig. 6-1.

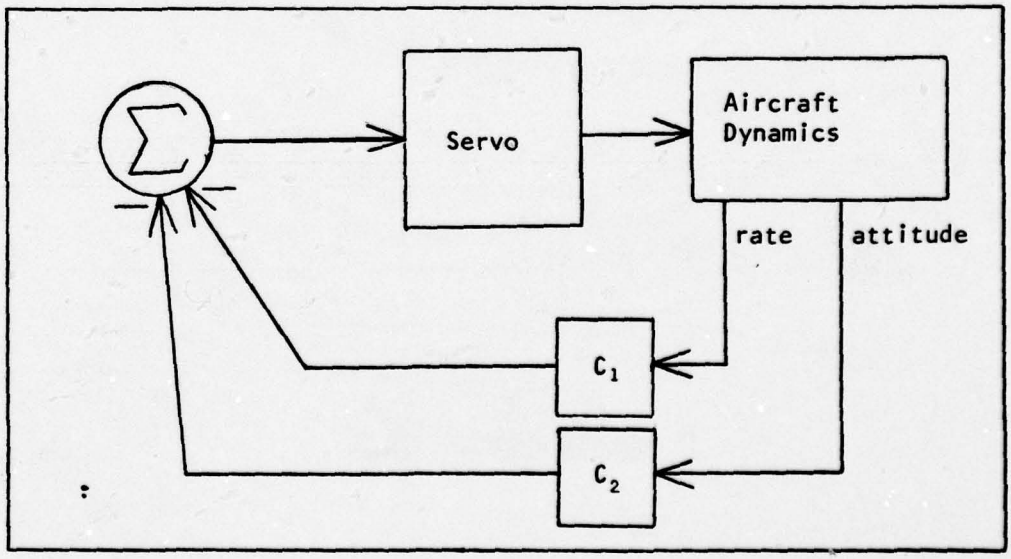


Figure 6-1: Basic Autopilot Block Diagram

The basic autopilot algorithm (exclusive of any filters) consists of multiplying the input to the inner loop (rate) by a constant (C_1), multiplying the input to the outer loop (attitude = att) by a constant (C_2), summing the two products, and outputting the negative of the sum. In equation form:

$$\text{Output} = -\left[C_1\ (\text{rate}) + C_2\ (\text{att})\right] \tag{6-1}$$

However, in the XBQM-106, all signals use a 2.5 volt zero reference, (0-5 volts, 2.5 volt center). Thus, the constants must be multiplied by the deviation from 2.5 volts, not the actual input voltage. This is accomplished by subtracting 2.5 volts from each input before multiplication, then adding 2.5 volts to the final sum:

$$\text{Output} = -\left[C_1\ (\text{rate} - 2.5) + C_2\ (\text{att} - 2.5)\right] + 2.5 \tag{6-2}$$

Since unsigned magnitude arithmetic was selected for use in the microprocessor (because the A/D output is unsigned magnitude), the terms must be rearranged to prevent negative intermediate results:

$$\begin{aligned}
\text{Output} &= -\left[C_1\ (\text{rate}) - C_1\ (2.5) + C_2\ (\text{att}) - C_2\ (2.5)\right] + 2.5 \\
&= -\left[C_1\ (\text{rate}) + C_2\ (\text{att}) - 2.5\ (C_1 + C_2)\right] + 2.5 \\
&= -\left[C_1\ (\text{rate}) + C_2\ (\text{att}) - 2.5\ (C_1 + C_2) + 2.5\right] + 5 \\
&= 5 - \left[C_1\ (\text{rate}) + C_2\ (\text{att}) - 2.5\ (C_1 + C_2 - 1)\right]
\end{aligned} \tag{6-3}$$

Thus, the term in brackets represents the uninverted output, subtracting it from 5 inverts it (provides negative feedback).

Since analog full scale is 5 volts and digital full scale is 256 (maximum value of 8 bit word = 2^8), the analog to digital conversion

essentially multiplies the inputs by $\frac{256}{5} = 51.2$ per volt. Conversely, the digital to analog conversion divides the output by 51.2. Thus, to maintain an equality, all terms of Eq (6-3) must be multiplied by 51.2:

$$\text{Output } (51.2) = 5\ (51.2) - \Big[C_1\ \{\text{rate} \cdot 51.2\} + C_2\ \{\text{att} \cdot 51.2\} - 2.5\ (C_1 + C_2 - 1)51.2\Big]$$

$$\text{Output } (51.2) = 256 - \Big[C_1\ \{\text{rate} \cdot 51.2\} + C_2\ \{\text{att} \cdot 51.2\} - 128\ (C_1 + C_2 - 1)\Big] \qquad (6\text{-}4)$$

Also, since fixed point arithmetic is used, a "pseudo floating point" method must be used. This is because all the input multiplication constants (loop gains) are in the range .3 to 2.7. Rounding to the nearest integer can result in excessive errors. For example, a constant of 1.4 would round to 1, an error of approximately 29%. The "pseudo floating point" method used to overcome this problem was to multiply all inputs by an additional constant, then divide the output by the same constant. The division is accomplished by shifting the binary point to the left, a very fast method of division. Obviously, the additional constant must be a power of 2 to enable division by binary point shifting. The larger the additional constant, the less rounding error that will occur. However, the size of the computer data word limits the maximum size. Eq (6-5) illustrates the use of the additional constant, 2^n, where n is an integer.

$$\{\text{Output } (51.2)\}\ 2^n = 256\ (2^n) - \Big[2^n\ (C_1)\ \{\text{rate } (51.2)\} + 2^n\ (C_2)\ \{\text{att } (51.2)\} - 128\ (C_1+C_2-1)\ 2^n\Big] \qquad (6\text{-}5)$$

The contents of the brackets consists of three terms, the first two of which are always positive, and the last one which is always negative (assuming $C_1 + C_2 > 1$, which is always the case in this application). Each of the first two terms consists of an eight-bit constant $(2^n) \cdot (C)$ times an eight-bit input (input)$\cdot$(51.2) which results in a sixteen-bit product. Two such products are added together, but the sum is still limited to sixteen bits. To determine the largest value of 2^n allowable, a worst case situation is considered in which both inputs are maximum. In this case:

$$\begin{aligned} \text{16-bit sum} &= 2^n\ (C_1)\ (256) + 2^n\ (C_2)\ (256) \\ &= 256\ (2^n)\ (C_1 + C_2) \end{aligned}$$

If we limit $C_1 + C_2 \leq 4$ (a reasonable assumption for this application), the largest value of 2^n can be determined. Since the largest value of a sixteen-bit word is $2^{16} = 65536$, then

$$\begin{aligned} 65536 &= 256\ (2^n)\ (4) \\ 2^n &= 64 \end{aligned}$$

which is an exact power of two. Substituting into Eq (6-5) yields:

$$\begin{aligned} \{\text{Output (51.2)}\}\ 64 = 256\ (64) - \Big[&64 \cdot C_1\ \{\text{rate (51.2)}\} \\ &+ 64 \cdot C_2\ \{\text{att (51.2)}\} - 8192\ (C_1 + C_2 - 1)\Big] \end{aligned}$$

In actual practice, the inversion process (subtraction of the term in brackets from 256 $\cdot$ 64) is accomplished after the binary point is shifted six places left to effect a division by 64.

$$\text{Output}(51.2) = 256 - \left\lfloor \frac{64C_1\{\text{rate}(51.2)\} + 64C_2\{\text{att}(51.2)\} - 8192(C_1+C_2-1)}{64} \right\rfloor$$

$$\text{Output} = \frac{256 - \left\lfloor \dfrac{64C_1\{\text{rate}(51.2)\} + 64C_2\{\text{att}(51.2)\} - 8192(C_1+C_2-1)}{64} \right\rfloor}{51.2} \qquad (6\text{-}6)$$

Note that division by 64 is accomplished by binary point shifting and the division by 51.2 is accomplished by the digital to analog conversion.

In actual use a rounded constant must be used. For example, if $C_1 = 1.7$, then $64C_1 = 108.8$, which rounds to 109. The C_1 that is actually used, called C_1', must then be calculated.

$$C_1' = \frac{109}{64} = 1.703125$$

The same procedure must be followed for C_2 to obtain C_2'. C_1' and C_2' are then used in Eq (6-6).

As previously noted, autopilot inputs and outputs are in the 2.5 ± 2.5 volt range. However, since some of the constants are greater than unity, the algorithm may calculate outputs greater than 5 volts or less than zero. For example, using Eq (6-6): If $C_1 = .5$, $C_2 = 2$, rate input = 2.5, and attitude input = 4, an output of -.5 is generated. Changing the attitude input to 1 results in an output of 5.5.

Since the I/O hardware is not capable of generating voltages outside the 0-5 volt range, limiters must be used to prevent erroneous results. Such limiters are implemented in software. Rearranging Eq (6-6) gives:

$$\lfloor 256 - \text{Output}(51.2) \rfloor\, 64 = 64C_1\{\text{rate}(51.2)\} + 64C_2\{\text{att}(51.2)\} - 8192(C_1+C_2-1)$$

Substituting the minimum and maximum output values, 0 and 5, into the left side, the right side of the equation is shown to be limited to the range 0 to 16384. The limiter is accomplished as follows:

If:

$$\left[64C_1\{rate(51.2)\}+64C_2\{att(51.2)\}-8192(C_1+C_2-1)\right] > 16384$$

$$64C_1\{rate(51.2)\}+64C_2\{att(51.2)\} - \left[16384+8192(C_1+C_2-1)\right] >0 \quad (6\text{-}7)$$

then output = 0.

Also, if

$$64C_1\{rate(51.2)\}+64C_2\{att(51.2)\}-8192(C_1+C_2-1) < 0 \quad (6\text{-}8)$$

then output = 5.

The flowchart in fig. 6-2 summarizes the single axis algorithm.

Filter Algorithm Development

So far, the autopilot algorithms have been described without regard to filters. However, the XBQM-106 autopilot contains a washout (high-pass) filter in the yaw axis. The basic digital algorithm (or linear difference equation) for this filter (described in Chapter IV) is:

$$Y(k) = A_o \cdot X(k) + A_1 \cdot X(k-1) + A_2 \cdot X(k-2) + B_1 \cdot Y(k-1) + B_2 \cdot Y(k-2) \quad (6\text{-}9)$$

where:

X(k) = current filter input
X(k-1) = filter input from previous iteration
X(k-2) = filter input from 2 iterations back
Y(k) = calculated filter output
Y(k-1) = calculated filter output from previous iteration
Y(k-2) = calculated filter output from 2 iterations back

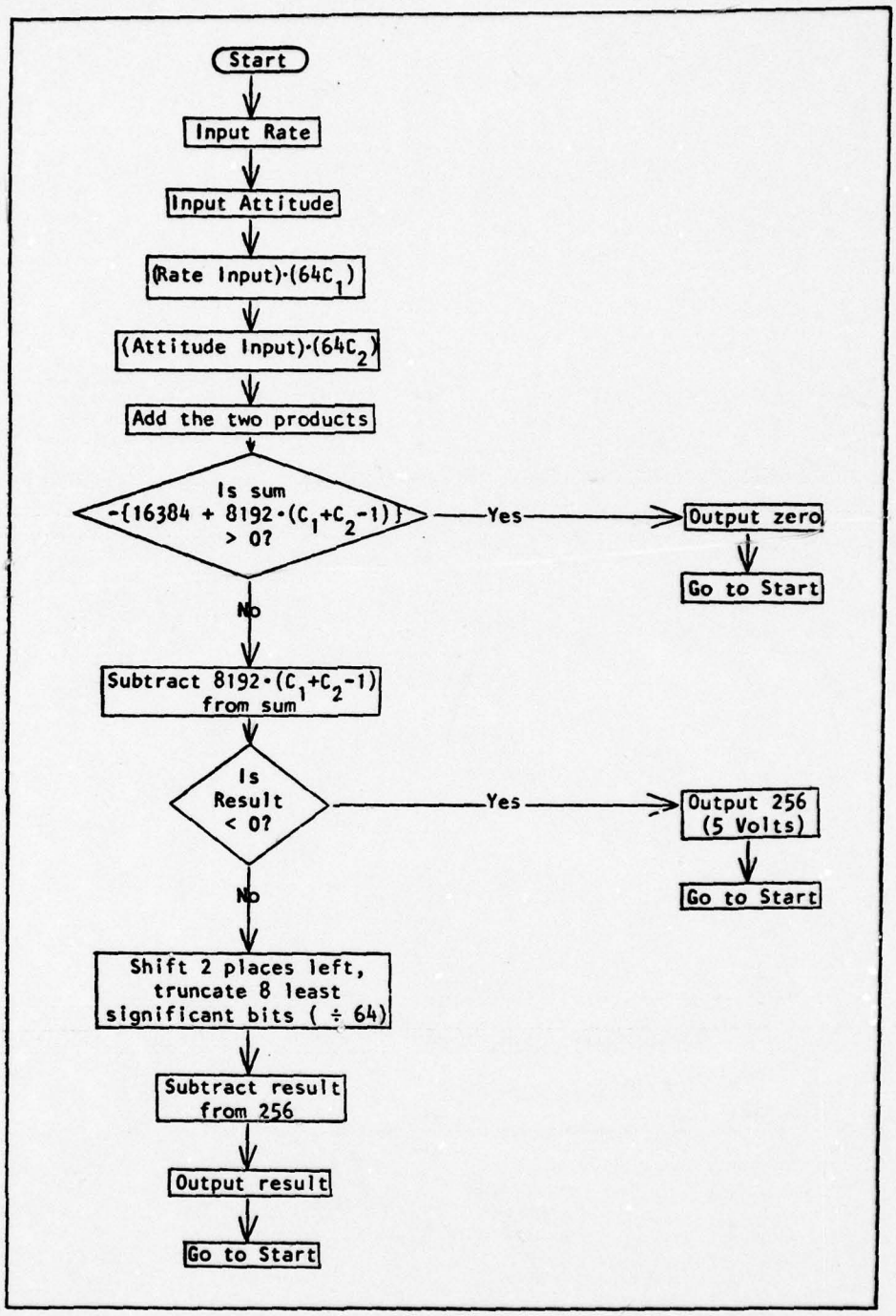


Figure 6-2: Single Axis Flowchart

Again, since the XBQM-106 uses a 2.5 volt zero reference (2.5 ± 2.5 V), the signals must mathematically be shifted to a zero volt zero reference, the calculations performed, then the output shifted back up 2.5 V.

As a result:

$$Y(k) = A_o\left[X(k)-2.5\right]+A_1\left[X(k-1)-2.5\right]+A_2\left[X(k-2)-2.5\right]+B_1\left[Y(k-1)-2.5\right] +B_2\left[Y(k-2)-2.5\right]+2.5$$

Rearranging to avoid negative intermediate results yields:

$$Y(k) = A_o \cdot X(k) + A_1 \cdot X(k-1) + A_2 \cdot X(k-2) + B_1 \cdot Y(k-1) + B_2 \cdot Y(k-2) -2.5(A_o+A_1+A_2+B_1+B_2-1) \qquad (6\text{-}10)$$

The actual values of the constants (obtained from Chapter IV) are as follows:

$$A_o = .98762018$$
$$A_1 = -1.9752404$$
$$A_2 = .98762018$$
$$B_1 = 1.9751712$$
$$B_2 = -.97530950$$

If these constants are rounded to the nearest integer, the values become 1, -2, 1, 2, -1, respectively. Since the filter characteristics are quite sensitive to the accuracy of the constants, an additional constant will again have to be employed to reduce roundoff error. Each term will be multiplied by the constant, then the final result divided by the constant to recover Y(k). The division is accomplished by binary

point shifting, thus the constant must again be of the form 2^n, where n is an integer.

$$Y(k)\cdot(2^n) = 2^n(A_o)X(k)+2^n(A_1)X(k-1)+2^n(A_2)X(k-2)+2^n(B_1)Y(k-1)$$
$$+2^n(B_2)Y(k-2)-2^n(2.5)(A_o+A_1+A_2+B_1+B_2-1)$$

The larger that 2^n is, the less roundoff error will occur; however, it should be small enough to avoid overflowing the sixteen-bit sums during intermediate steps. Since unsigned magnitude arithmetic is used, the terms with positive constants must be added first to avoid negative sums. Thus, positive terms will determine the maximum value of 2^n.

$$2^n(A_o)\ X(k) + 2^n(A_2)\ X(k-2) + 2^n(B_1)\ Y(k-1) \leq 65536$$

Using a worst case of 256 for X(k), X(k-2) and Y(k-1), and substituting for A_o, A_2 and B_1:

$$2^n(.98762018)(256)+2^n(.98762018)(256)+2^n(1.9851712)(256) \leq 65536$$
$$2^n(1011.305359) \leq 65536$$
$$2^n \leq 64.80337459$$

Therefore, let $2^n = 64$, which yields

$$Y(k)\cdot(64) = 64(A_o)X(k)+64(A_1)X(k-1)+64(A_2)X(k-2)+64(B_1)Y(k-1)$$
$$+64(B_2)Y(k-2)-64(2.5)(A_o+A_1+A_2+B_1+B_2-1) \qquad (6\text{-}11)$$

Multiplying each constant by 64 and rounding yields:

$$A_o : \quad .98762018\ (64) = \quad 63.20769152 \rightarrow \quad 63$$
$$A_1 : -1.9752404\ \ (64) = -126.4153856 \rightarrow -126$$
$$A_2 : \quad .98762018\ (64) = \quad 63.20769152 \rightarrow \quad 63$$
$$B_1 : \quad 1.9751712\ \ (64) = \quad 126.4109569 \rightarrow 126$$
$$B_2 : -.97530950\ (64) = -62.419808 \rightarrow -62$$

Dividing each rounded result by 64 yields the actual value of the constant used.

$$A_o' = \frac{63}{64} = .984375$$
$$A_1' = \frac{-126}{64} = -1.96875$$
$$A_2' = \frac{63}{64} = .984375$$
$$B_1' = \frac{126}{64} = 1.96875$$
$$B_2' = \frac{-62}{64} = -.96875$$

Substituting these values in Eq (6-11) yields:

$$Y(k)\cdot(64) = 63\cdot X(k) - 126\cdot X(k-1) + 63\cdot X(k-2) + 126\cdot Y(k-1) - 62\cdot Y(k-2)$$
$$-64(2.5)(.984375-1.96875+.984375+1.96875-.96875-1)$$

$$Y(k)\cdot(64) = 63\cdot X(k) - 126\cdot X(k-1) + 63\cdot X(k-2) + 126\cdot Y(k-1) - 62\cdot Y(k-2) - 0$$

$$Y(k) = \frac{63\cdot X(k) + 63\cdot X(k-2) + 126\cdot Y(k-1) - 126\cdot X(k-1) - 62\cdot Y(k-2)}{64} \qquad (6\text{-}12)$$

The division by 64 is accomplished by binary point shifting.

While the inputs are eight bits, the intermediate sums and filter output are sixteen bits (only eight bits are output from the filter algorithm; however, sixteen bits are available internal to the algorithm). It was found experimentally that eight bit accuracy of Y(k-1) and Y(k-2) was not sufficient to provide proper filter results (see explanation in Chapter IV). The solution was to break Y(k-1) and Y(k-2) into two terms, integer and fraction, to achieve sixteen bit accuracy in these terms. Since the fractional part is stored as an integer number, the product of the integer portion times a constant must be divided by eight (using binary point shifting) before inclusion in the summation.

$$Y(k) = \frac{\left[63\cdot X(k) + 63\cdot X(k-2) + 126\cdot Y(k-1)_{int} + \{126\cdot Y(k-1)_{frac} \div 8\} - 126\cdot X(k-1) - 62\cdot Y(k-2)_{int} - \{62\cdot Y(k-2)_{frac} \div 8\}\right]}{64} \qquad (6\text{-}13)$$

Another situation that requires additional consideration is the filter response to a step input. A step of sufficient magnitude (depends on prior inputs and outputs, but always greater than 2.5 volts) will produce overshoot on the output. In many cases, this presents no problem; however, in certain cases the overshoot will carry outside the 0-5 volt output range (0-256 internally). Since the hardware cannot generate such values, the algorithm enters an erroneous and irrecoverable state. To prevent such an occurrence, a software check is made to ensure that a negative result does not occur after any of the subtractions, and that the final sum (before division by 64) does not exceed 16383

($256(64) - 1 = 3FFF_{16}$). If such a condition occurs, X(k-1), X(k-2), Y(k-1), and Y(k-2) are reinitialized to 128 to make it "look" like the step origin was from 128 (2.5 V), then Y(k) is recalculated. This will correct any situation since the problem cannot occur whenever the step magnitude $\leq$ 128. The conditions leading to this problem were examined by executing the algorithm on a HP-29C programmable calculator, one step at a time. This allowed for analysis of the situation and formulation of a recovery technique. As a by-product, the algorithm does not need initialization; it is self-initializing whenever erroneous results occur. The HP-29C program is listed in Appendix D. Fig. 6-3 lists the complete filter algorithm.

3-Axis Algorithm

In the 3-axis autopilot program, the filter routine is interleaved with the data inputs. The A/D conversions are accomplished in parallel with the filter routine, interrupt I/O being used to input data when A/D conversion is complete. This technique is possible because A/D conversions take approximately 220 μsec (could be decreased to 100 μsec, if necessary) while each software multiplication takes from 650 to 800 μsec (determined by number of instructions x instruction time). Fig. 6-4 is a flowchart of the entire 3-axis autopilot program. The actual program listing is in Appendix E . The values of constants used are listed in fig. 6-5, while the CPU register assignments are listed in fig. 6-6.

The complete 3-axis program also contains three subroutines (add, subtract, and multiply), and an interrupt routine to input data.

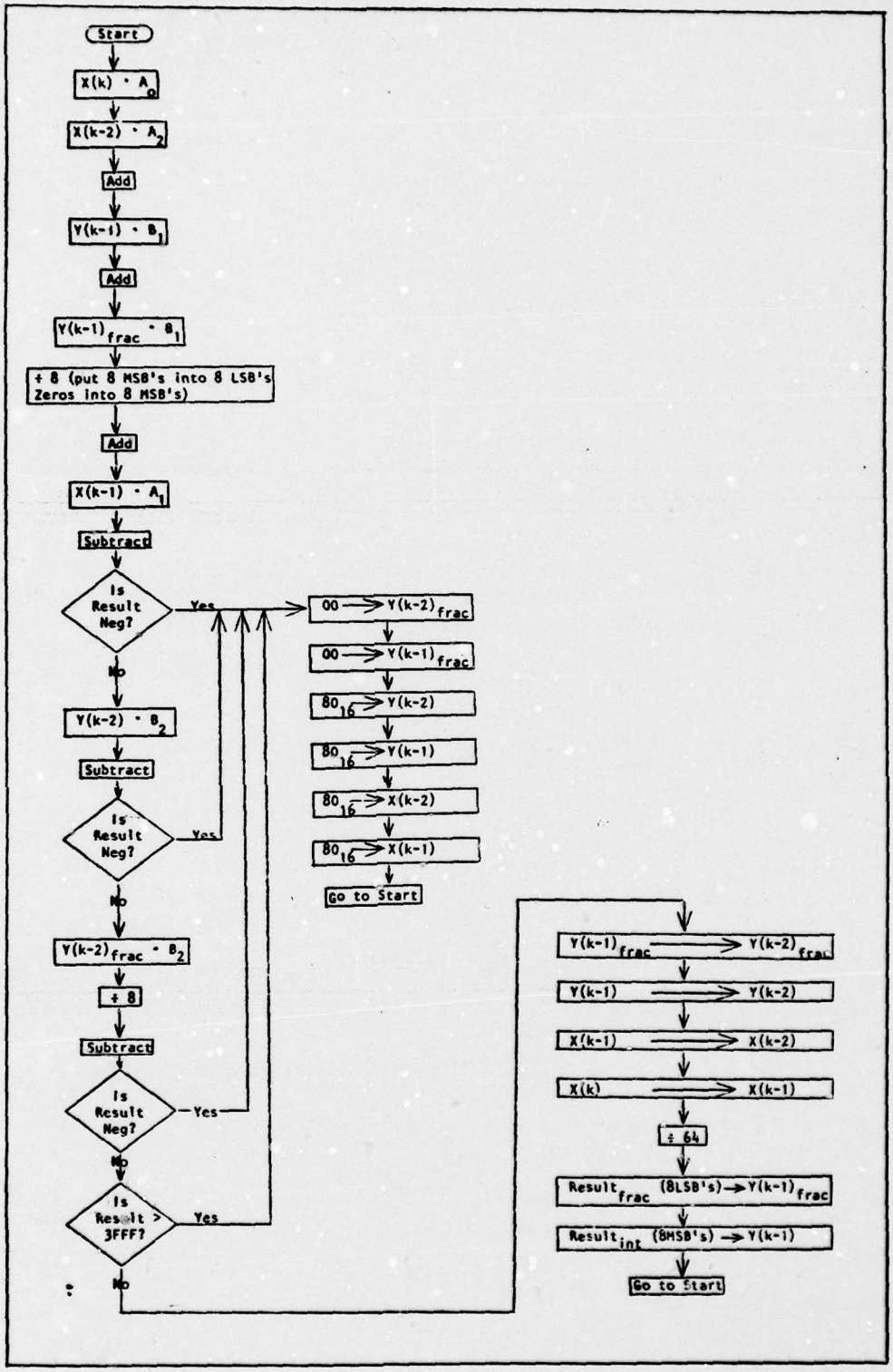


Figure 6-3: Filter Flowchart

Figs. 6-7 through 6-10 are flowcharts for these routines.

Chapter VII will describe how this software, along with the hardware described in chapter V, was used in conjunction with a hybrid computer simulation to test the performance of the entire concept.

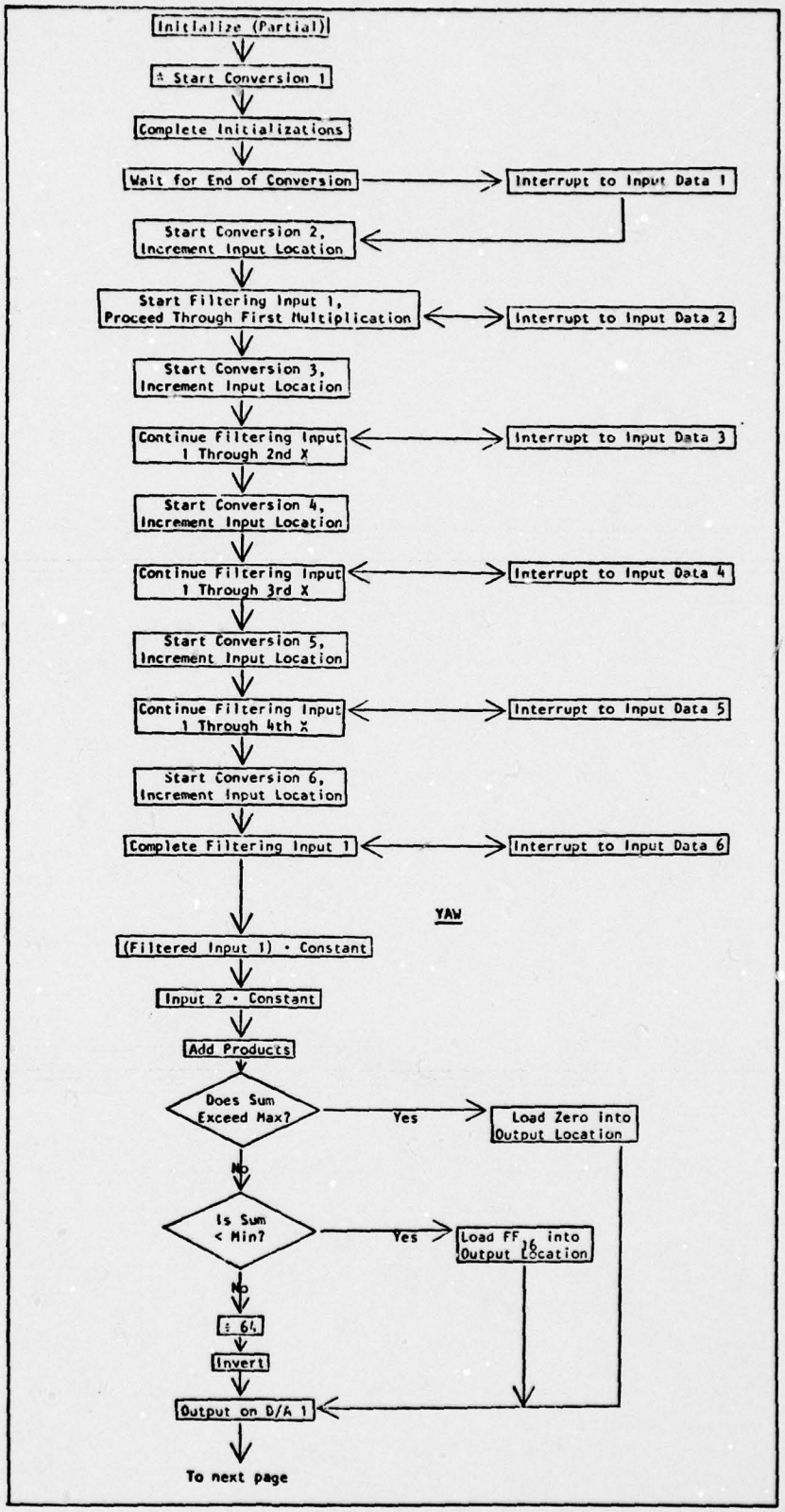


Figure 6-4: 3-Axis Flowchart

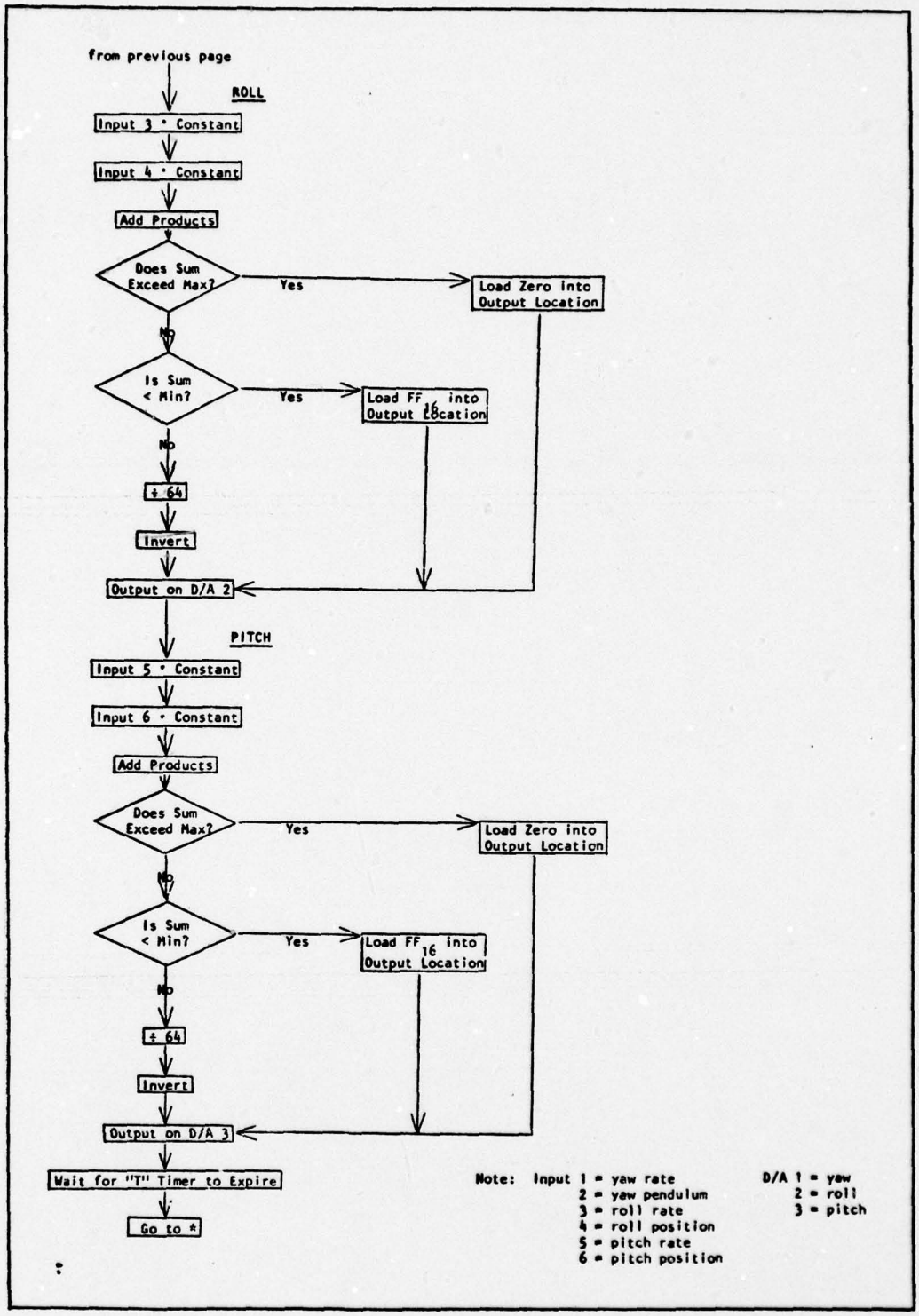


Figure 6-4: 3-Axis Flowchart (continued)

Axis	C_1	$64C_1$	$64C_1$ Rounded	C_1'	C_2	$64C_2$	$64C_2$ Rounded	C_2'	$8192\ (C_1'+C_2'-1)$	$16384 + 8192\ (C_1'+C_2'-1)$
pitch	$K_\theta = 2.546$	162.944	$163 = A3_{16}$	$\frac{163}{64} = 2.546875$	$K_{\dot{\theta}} = .445$	28.480	$28 = 1C_{16}$	$\frac{28}{64} = .437500$	$16256 = 3F80_{16}$	$32639 = 7F7F_{16}$
roll	$K_\phi = 1.588$	101.632	$102 = 66_{16}$	$\frac{102}{64} = 1.593750$	$K_{\dot{\phi}} = .452$	28.928	$29 = 1D_{16}$	$\frac{29}{64} = .453125$	$8576 = 2180_{16}$	$24959 = 617F_{16}$
yaw	$K_\psi = 1.830$	117.120	$117 = 75_{16}$	$\frac{117}{64} = 1.828125$	$K_{a_y} = .363$	23.232	$23 = 17_{16}$	$\frac{23}{64} = .359375$	$9728 = 2600_{16}$	$26111 = 65FF_{16}$

Figure 6-5: Autopilot Constant Calculations
(for use in Eqs. 6-6, 6-7, and 6-8)

R0	Program counter
R1	Interrupt routine pointer
R2	Stack pointer
R3	- Multiply subroutine operand 1 pointer and result upper byte pointer - Add and subtract subroutines operand 1 pointer
R4	- Multiply subroutine operand 2 pointer and result lower byte pointer - Add and subtract subroutines operand 2 pointer
R5	Used by multiply subroutine for accumulated product
R6.0	Multiply subroutine loop counter
R6	Pointer to ALE address location
R7	Multiply subroutine program counter
R8	Subtract subroutine program counter
R9	Add subroutine program counter
RA	Pointer for input data
RB-RF	Not used

Figure 6-6: CPU Register Assignments

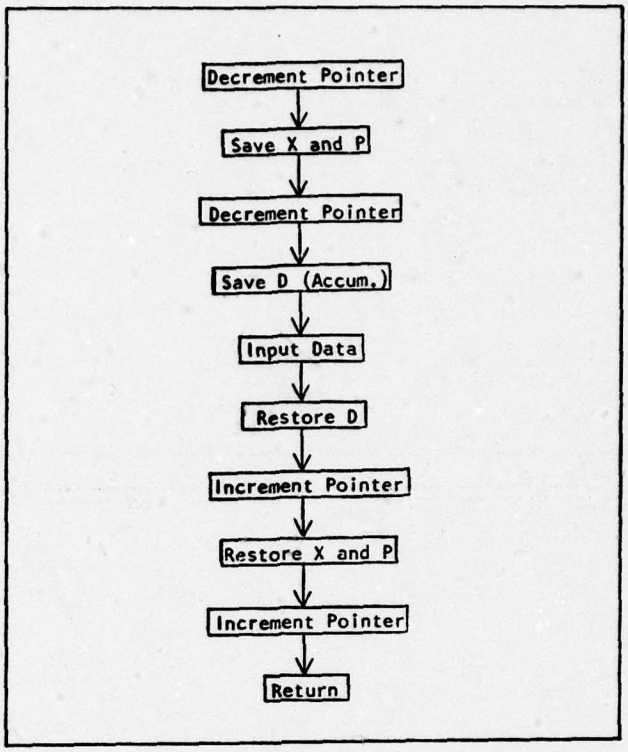


Figure 6-7: Interrupt Routine to Input Data Flowchart

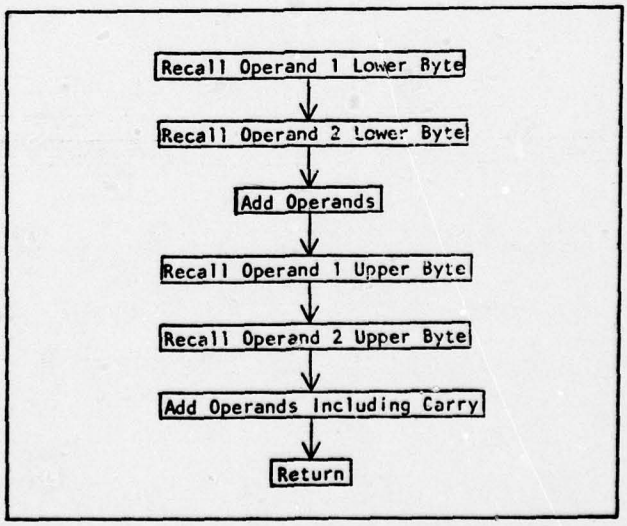


Figure 6-8: Add Subroutine Flowchart

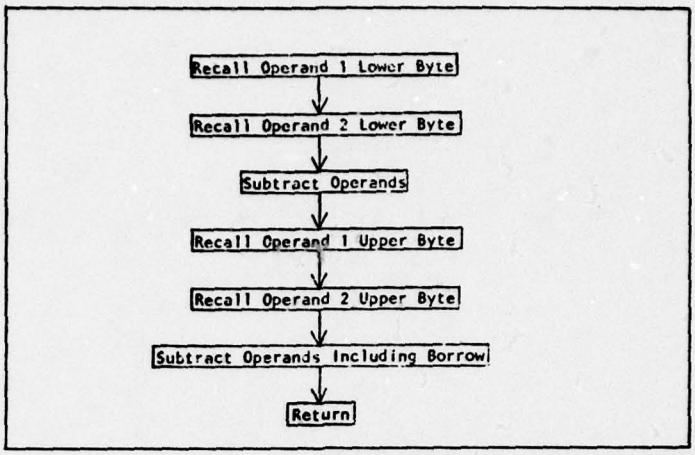


Figure 6-9: Subtract Subroutine Flowchart

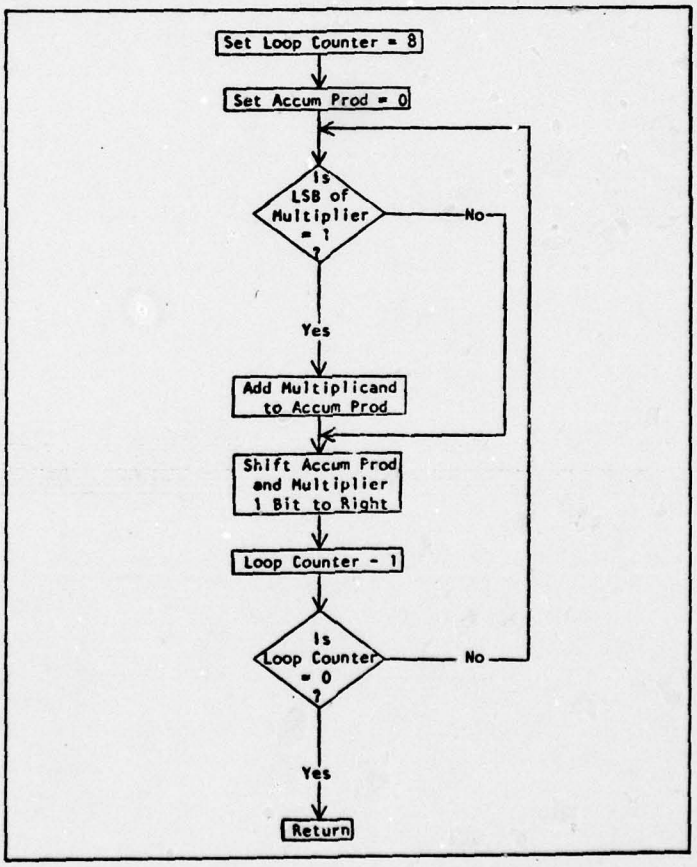


Figure 6-10: Multiply Subroutine Flowchart

VII. Test Procedures and Results

After development of the required hardware and software, the resulting system was tested for proper operation. Testing was accomplished in several stages of increasing complexity. The results were then evaluated to determine the performance of the digital A/P, and thus, a measure of success towards meeting the study objectives.

Test Procedures

Testing was accomplished in two basic modes: static and dynamic. Static testing involves inputting signals from a DC power supply and/or AC function generator, and observing the output on an oscilloscope and/or chart recorder. This mode requires no special equipment and can be accomplished in the School of Engineering laboratories. The intent of this method is to verify the basic operation of the algorithm before proceeding to more sophisticated testing. Such items as gain values, limiting processes, filter operation, and frequency limits can be observed in this manner.

Dynamic testing involves interfacing with a six degree of freedom hybrid simulation of the XBQM-106 aircraft, sensors, actuators, and analog A/P that is resident in the WPAFB computer center. The time responses to various inputs can be observed in this manner, and the digital A/P performance can be compared to the simulation of the analog A/P. (As a by-product, several errors in the hybrid simulation were corrected when the digital A/P response failed to agree with the analog simulation.) In order to facilitate quick hook-up to the simulation patch board, a special cable and plug-in printed circuit card were fabricated to allow

interfacing by simply plugging the card into the prototyping board.

A variation of the dynamic testing mode is the use of a mock-up control station to input pilot commands to the system. Fig. 7-1 shows the digital A/P connected to the hybrid simulation, and Fig. 7-2 shows the control station mock-up.

Testing is accomplished in order of increasing complexity. The following is a chronological list of the testing process:

Static Single-Axis Testing. This test was used mainly as a software debugging aid. Once the algorithm appeared to be operating properly, the next step could be accomplished.

Dynamic Single-Axis Testing. The initial attempt of this method was to verify proper hook-up to the hybrid simulation. Once accomplished, time responses to various inputs were obtained. Comparison to the analog A/P response allowed verification of the A/P loop gains. The single-axis static and dynamic tests were then repeated for each of the other two axes. Since the washout filter was not yet incorporated into the yaw axis, it was disconnected in the analog simulation.

Static Filter Testing. This method was used to assist in debugging the yaw washout filter software. Once proper operation was obtained, the filter frequency response characteristics were recorded on a chart recorder, and are discussed in the results section of this chapter.

Static Single-Axis w/Filter Testing. The yaw axis software and the filter software were combined and debugged.

Dynamic Single-Axis w/Filter Testing. The yaw filter in the analog simulation was reconnected and the digital A/P response compared to the

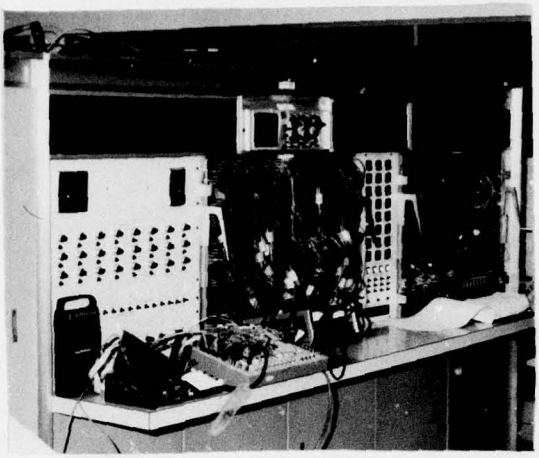


Figure 7-1: Digital Autopilot and Hybrid Simulation

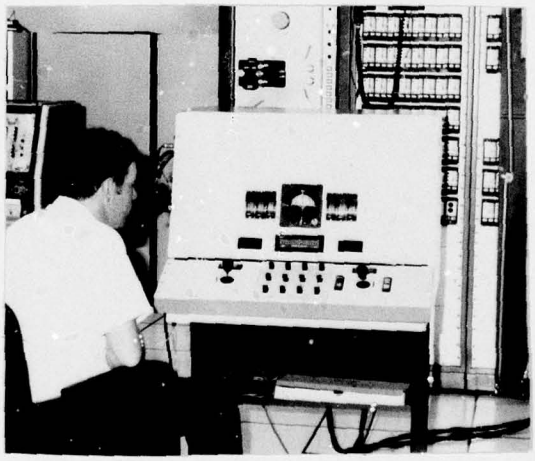


Figure 7-2: RPV Manual Control Station

analog simulation.

Static Three-Axis (w/Filter in Yaw) Testing. The software for the complete A/P was developed and debugged.

Dynamic Three-Axis (w/Filter in Yaw) Testing. The digital A/P responses were compared to the analog simulation, applying forcing functions to each axis separately, then all three axes simultaneously. Axis cross-coupling effects were observed. Time responses were recorded on chart recorders and are discussed in the results section of this chapter.

Dynamic Three-Axis (w/Filter in Yaw) Testing Using Command Console. The command console was used to determine which of the effects observed in the time response are significant. Personnel from the Mini-RPV Group (including the chief flight test pilot) were utilized to provide subjective judgements as to the "handling qualities" of the digital A/P compared to the analog A/P.

Results

Digital Washout Filter Testing Results. The digital washout (high-pass) filter in the yaw rate loop was implemented as a recursive software algorithm in the microprocessor. To insure numerical stability and desired performance, a frequency response test of the filter was accomplished. Sinusoids of varying frequencies were input to the filter, and output responses were recorded. Figure 7-3 gives the strip-chart results. Inputs are shown on the top, and outputs on the bottom of each run. Sampling frequency and the filter equations are based upon f_s = 50 HZ. Note that high-pass filtering is shown by the characteristic attenuation of the input sinusoidal amplitudes as frequencies decrease past the cutoff point, f_c = .25 HZ (1.57 rad/sec). This experimental cutoff frequency represents a 12% difference when compared to the infinite word length digital filter, f_c, in chapter IV (f_c = 1.4 rad/sec). This difference can be attributed to the effects of sixteen-bit word length arithmetic and possible errors in the recording equipment. Although not shown, phase coherence is also preserved.

Filter responses to input square-waves were also done in a similar manner. Results are shown in fig. 7-4. Filter performance is again demonstrated by noting that high frequency rise time components pass through the filter unattenuated, whereas the DC plateaus of the square-wave decay back to the 2.5 V center reference.

In general, this testing process provides an excellent means of "debugging" problems in the filter. Filter mechanization and proof of performance are achieved. Once the filter was shown to perform satisfactorily, it was incorporated in the yaw rate software control algorithm.

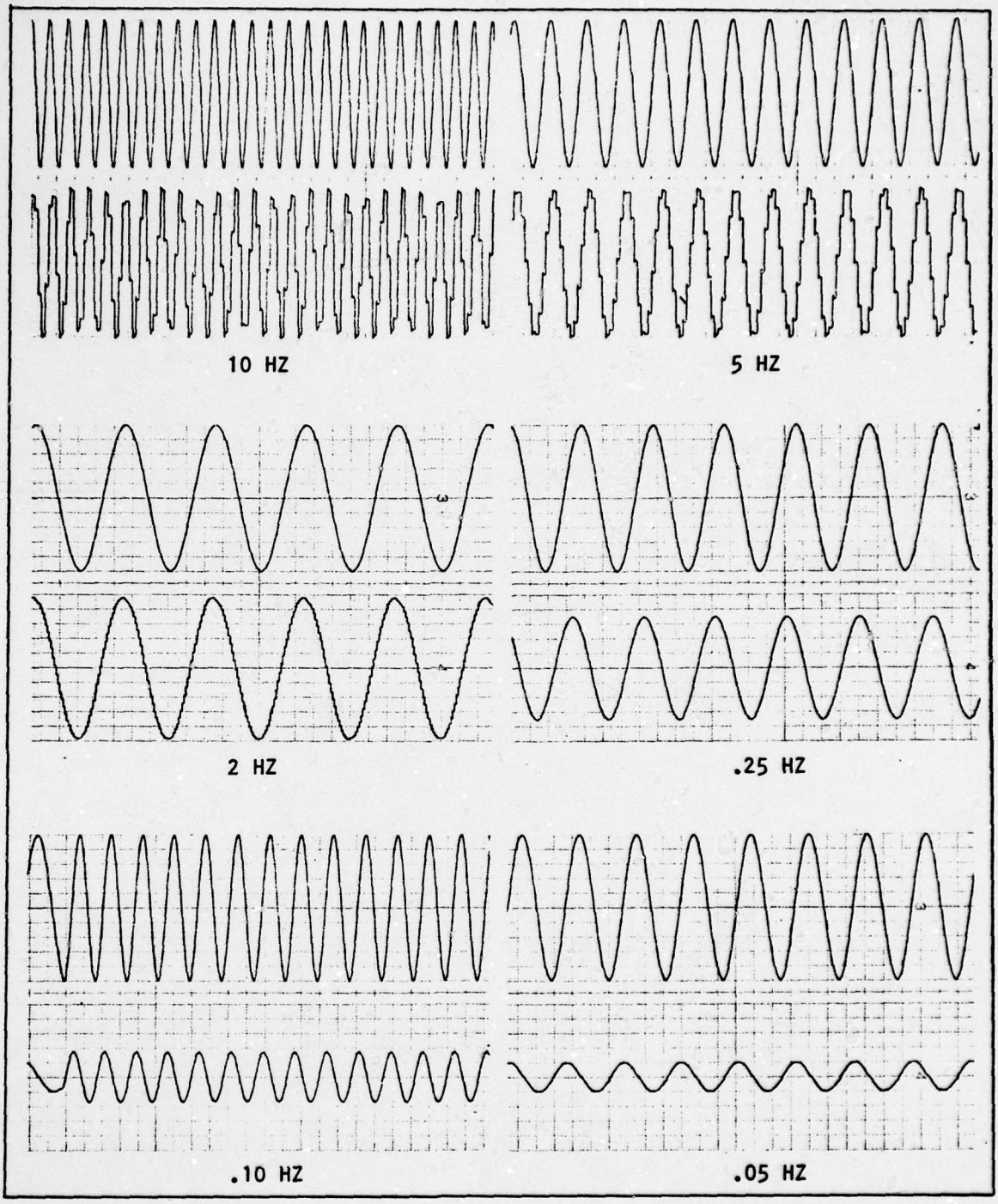


Figure 7-3: Frequency Response of Digital Filter (Sinusoid)

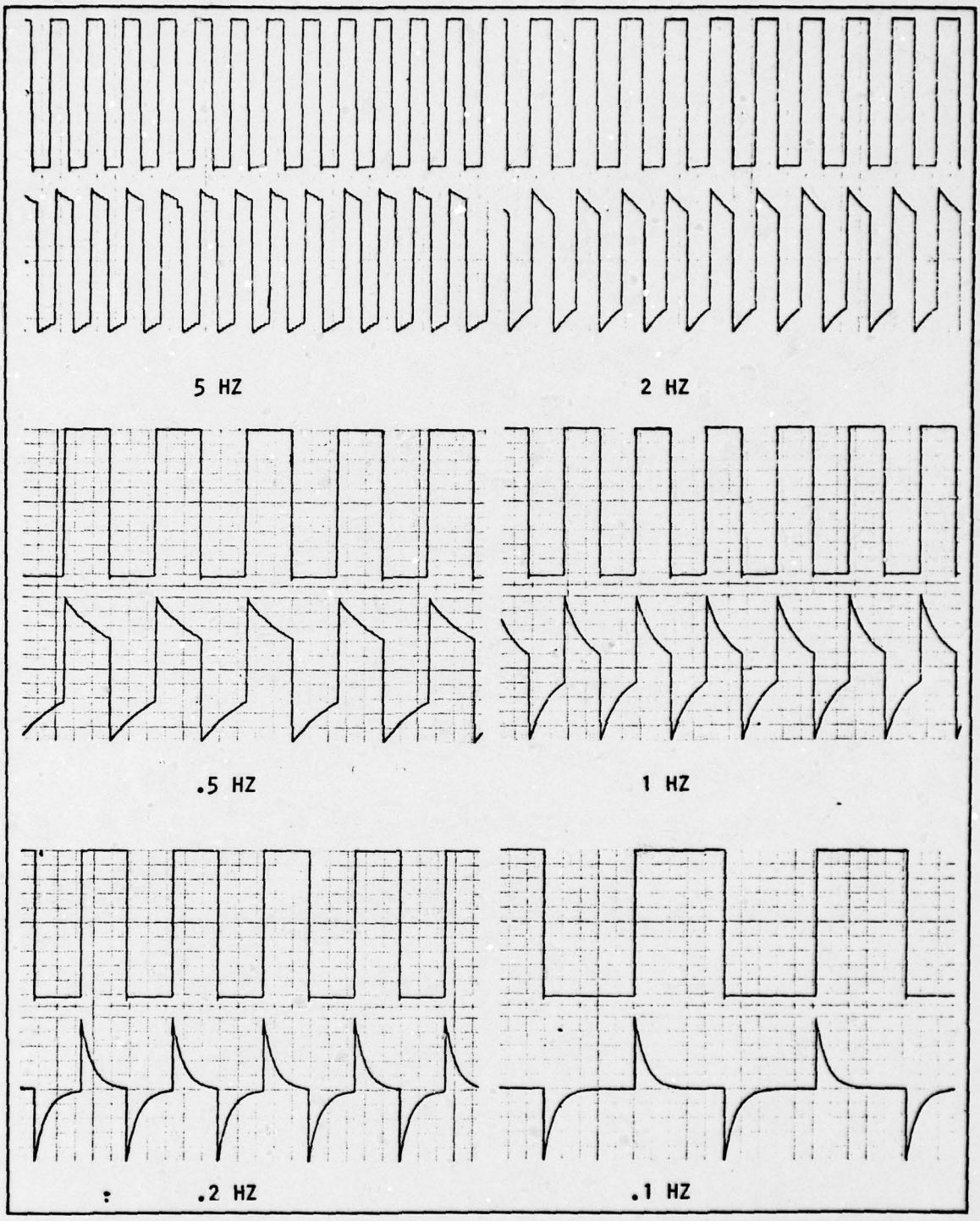


Figure 7-4: Frequency Response of Digital Filter (Square Wave)

Digital Versus Analog Autopilot Testing Results. The primary means of determining digital A/P performance is through a qualitative comparison of the time responses generated by each A/P (simulated analog versus digital). If similar performance characteristics are obtained, the objective of this study is met: to design a digital A/P capable of emulating the analog A/P. The analog A/P is represented on a hybrid simulator as a non-linear 6DOF model. Comparison with the actual flying A/P cannot be made at this time. The following is a summary of test results:

Pitch Attitude Controller. Figs. 7-5 and 7-6 present the analog and digital A/P responses, respectively, to a 20 degree step pitch up command (θ_c). The primary outputs of interest, pitch attitude (θ), pitch rate ($\dot{\theta}$), angle of attack (α), and elevator deflection (δ_e) show excellent reproduction. Both A/P's achieve the commanded 20 degree pitch angle in roughly the same time. Both drift down to the 12 degree steady-state attitude similarly. The digital A/P does appear to be somewhat less damped in the transient response, more so in $\dot{\theta}$. This may be attributed to quantization effects in the control software. Overall performance of this controller is quite comparable. No deleterious effects are noted.

Roll Attitude Controller. Figs. 7-7 and 7-8 provide the test results of the roll attitude controller to a 20 degree step roll right command (ϕ_c). The primary outputs of interest are roll angle (ϕ), roll rate ($\dot{\phi}$), sideslip angle (β), and aileron deflection (δ_a). Secondary coupled axis outputs include yaw angle (ψ), and yaw rate ($\dot{\psi}$). A comparison of the two A/P responses in ϕ reveals nearly identical waveshapes, both in the transient and steady-state. The digital $\dot{\phi}$ and δ_a responses show

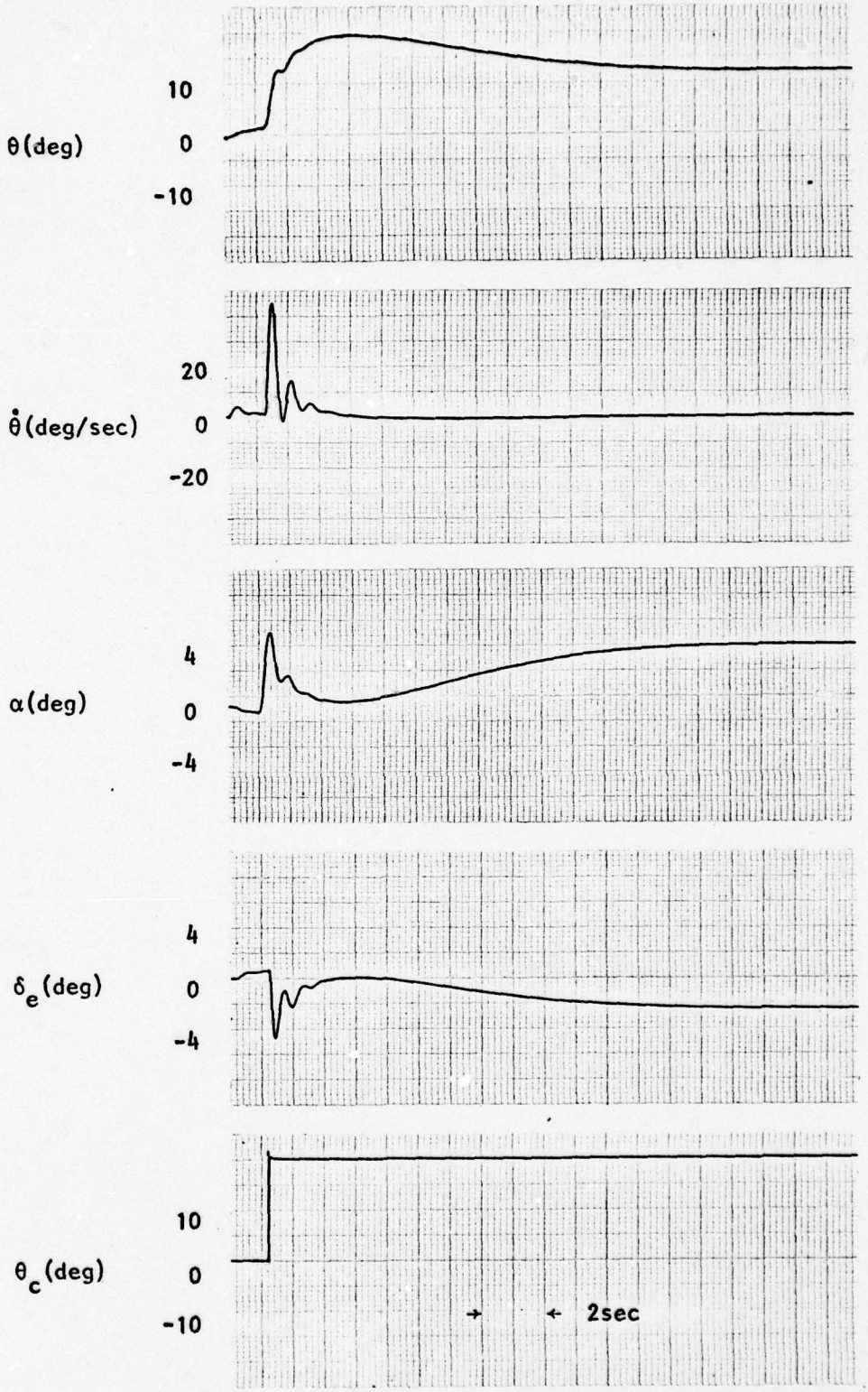


Figure 7-5 : RPV Response to 20 Degree Step Pitch Up Command(θ_c); Pitch Attitude/Rate Loops Closed; Analog Controller Engaged

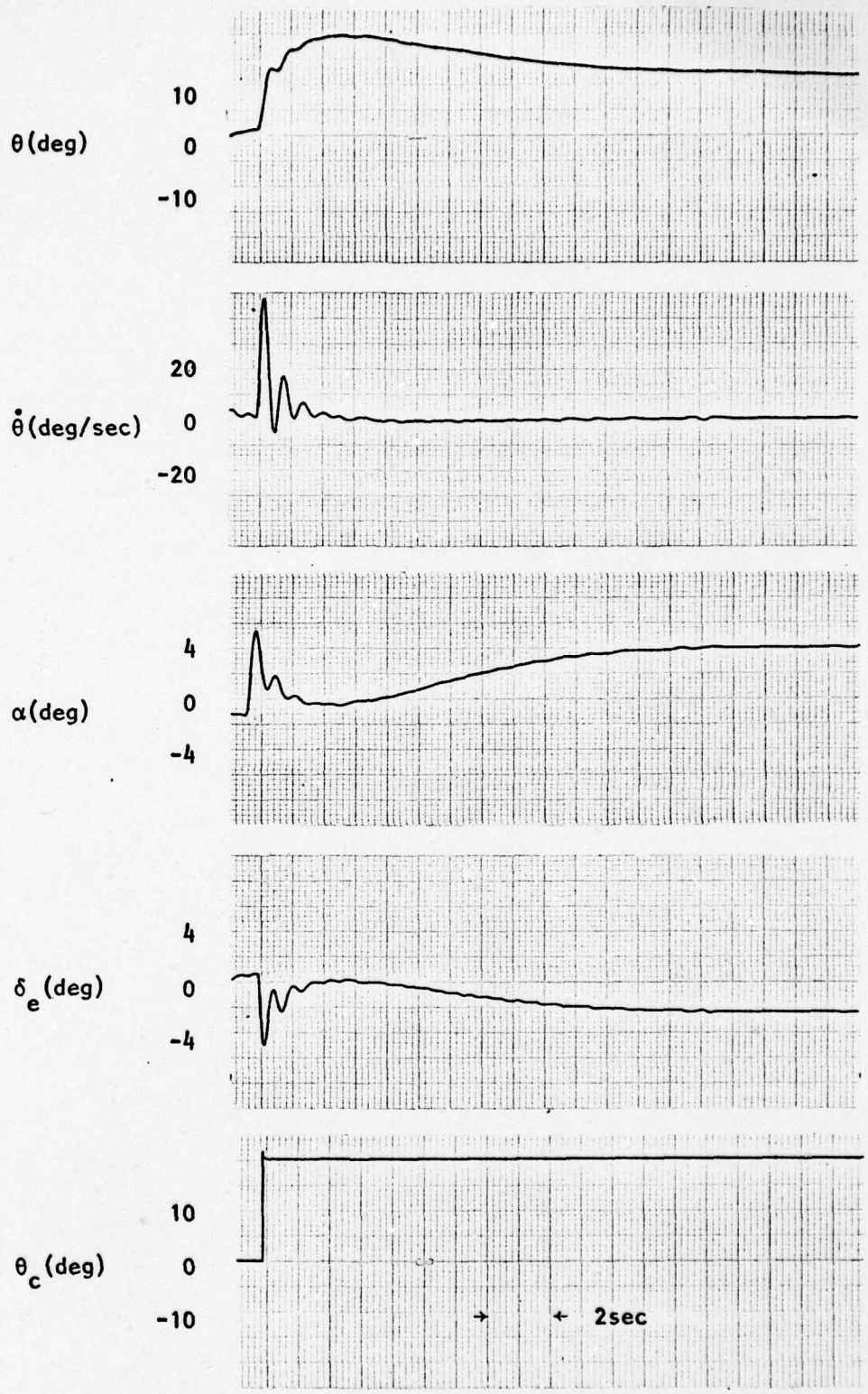


Figure 7-6 : RPV Response to 20 Degree Step Pitch Up Command(θ_c); Pitch Attitude/Rate Loops Closed; Digital Controller Engaged; f_s=50 Hz

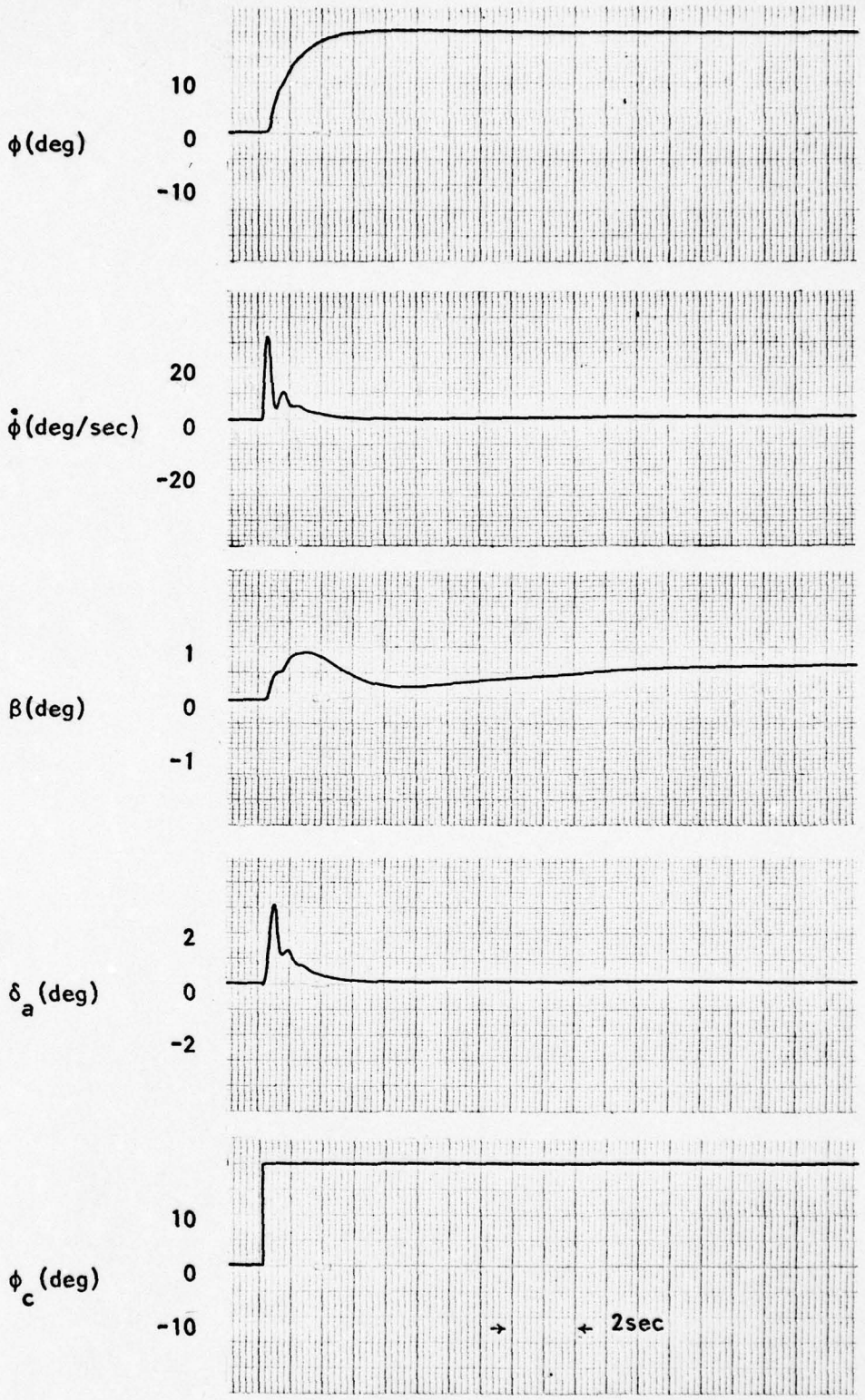


Figure 7-7 : RPV Response to 20 Degree Step Roll Right Command(ϕ_c); Roll Attitude/Rate Loops Closed; Analog Controller Engaged

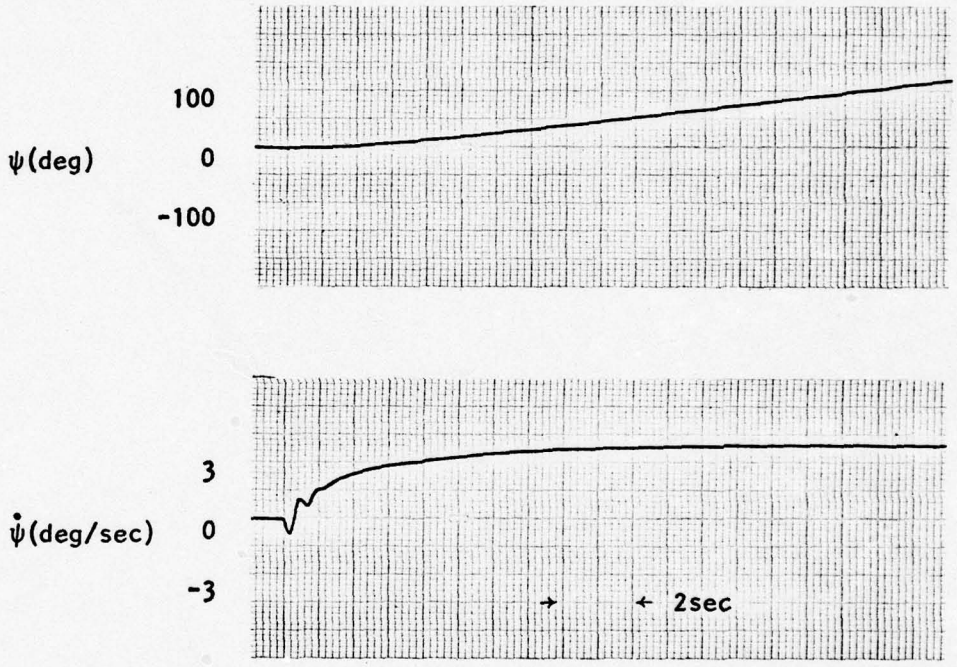


Figure 7-7 : Continued (Analog Lateral Controller Engaged)

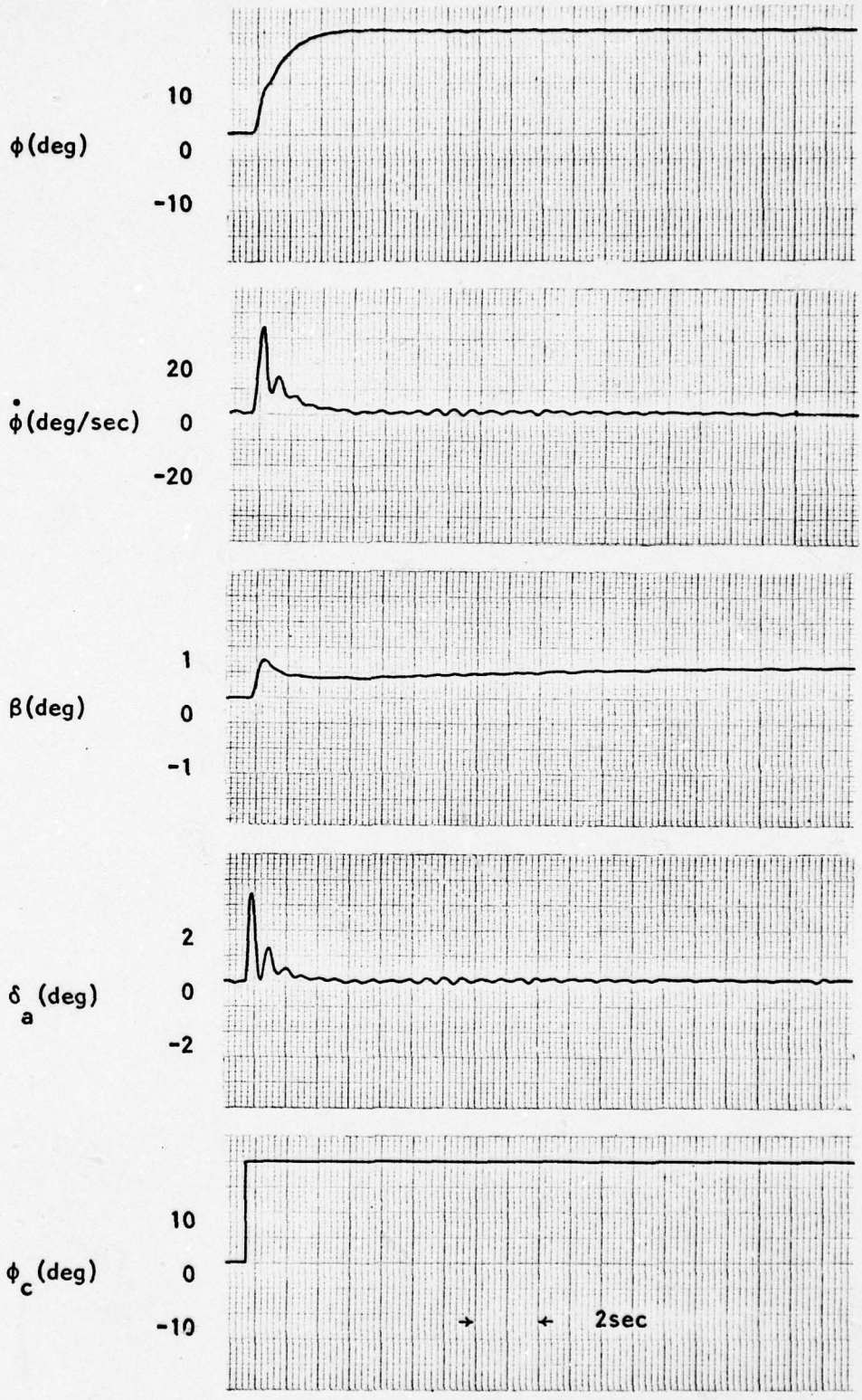


Figure 7-8 : RPV Response to 20 Degree Step Roll Right Command(ϕ_c); Roll Attitude/Rate Loops Closed; Digital Controller Engaged; f_s=50 Hz

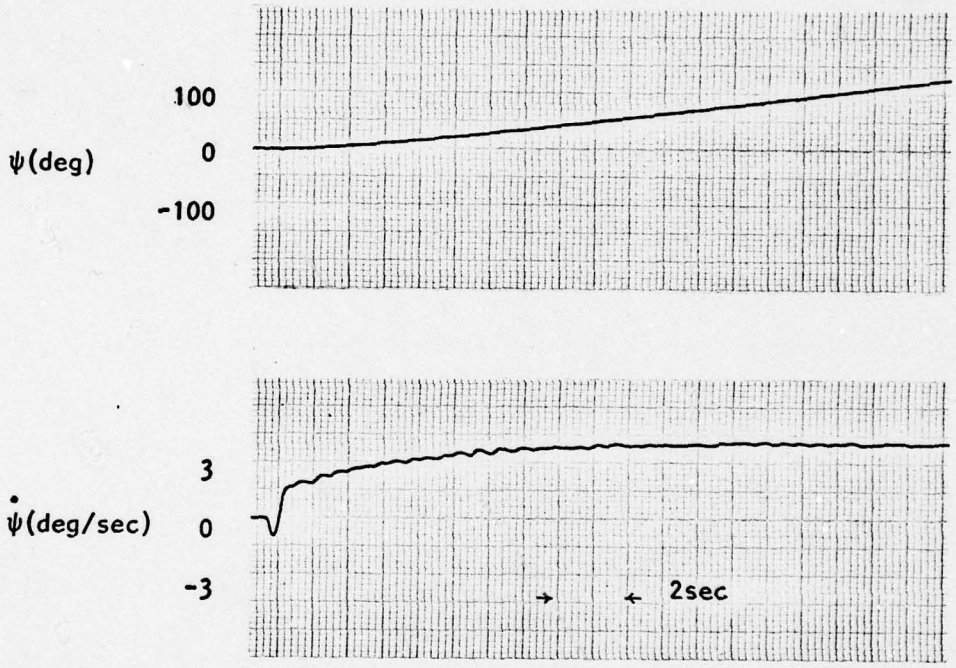


Figure 7-8 : Continued (Digital Lateral Controller Engaged; f_s=50 Hz)

slightly increased, but not excessive, rippling after 1.5 seconds. Again, finite wordlength and quantization effects are dominant. Performance, however, is not degraded due to the small peak-to-peak amplitudes of the oscillations. These similar oscillations are noted in digital $\dot{\psi}$ but are no problem. Results for this controller show good reproducibility.

Yaw Rate Controller. Figs. 7-9 and 7-10 illustrate the resulting responses of the yaw rate controller to a -3.3 deg/sec step yaw rate command ($\dot{\psi}_c$). Primary outputs include ψ, $\dot{\psi}$, β and rudder deflection (δ_r). Secondary cross-coupling outputs include ϕ, $\dot{\phi}$, and δ_a. ψ, $\dot{\psi}$ and β responses for both A/P's are nearly identical. $\dot{\psi}$ is shown to achieve the commanded yaw rate in the steady-state for both A/P's. A slight aberration is noted in the transient response of the digital rudder output (δ_r), but this should not be a problem.

A comparison of cross-coupling outputs, especially in roll rate ($\dot{\phi}$) and aileron deflection (δ_a) yield what first appears to be poor correlation. The digital outputs of $\dot{\phi}$ and δ_a are shown to go into sustained steady-state oscillations. At first glance, this behavior is unacceptable. On closer scrutiny, however, the results are acceptable. The peak-to-peak magnitudes of the oscillations are smaller than the resolution of the rate sensors and actuators. Thus, the RPV would not see these effects. Quantization and limit cycling may contribute to this phenomena. For simulation purposes, the yaw rate input to the rudder produces a cross-coupled sustained aileron (δ_a) oscillation of about 0.3 degrees. This is almost unnoticeable. When the digital A/P was flown on the manual control station, the oscillations could not be detected in any manner.

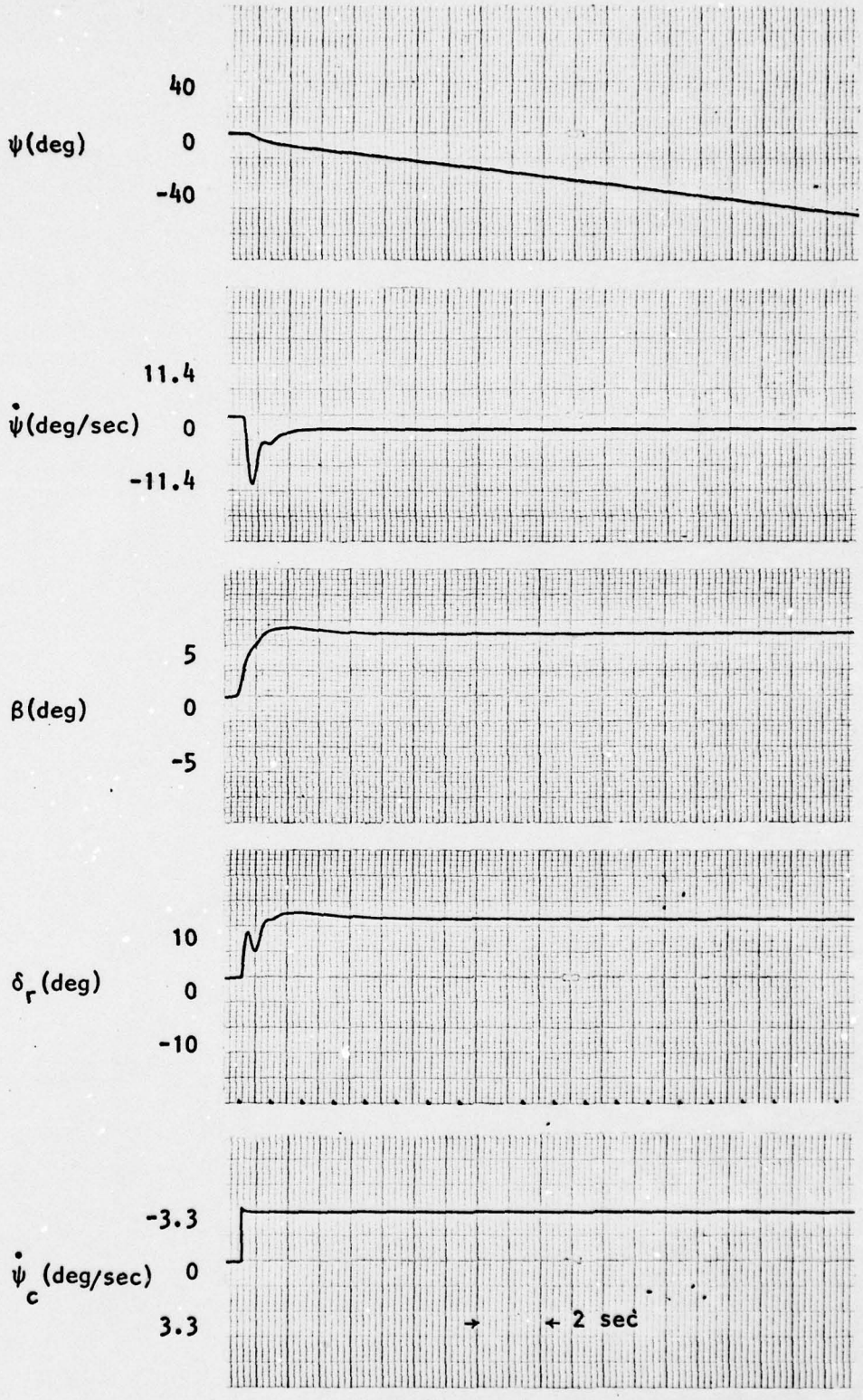


Figure 7-9 : RPV Response to 3.3 Deg/Sec Step Yaw Rate Command($\dot{\psi}$); Yaw Rate/Accelerometer Loops Closed; Analog Controller Engaged (Yaw Damper)

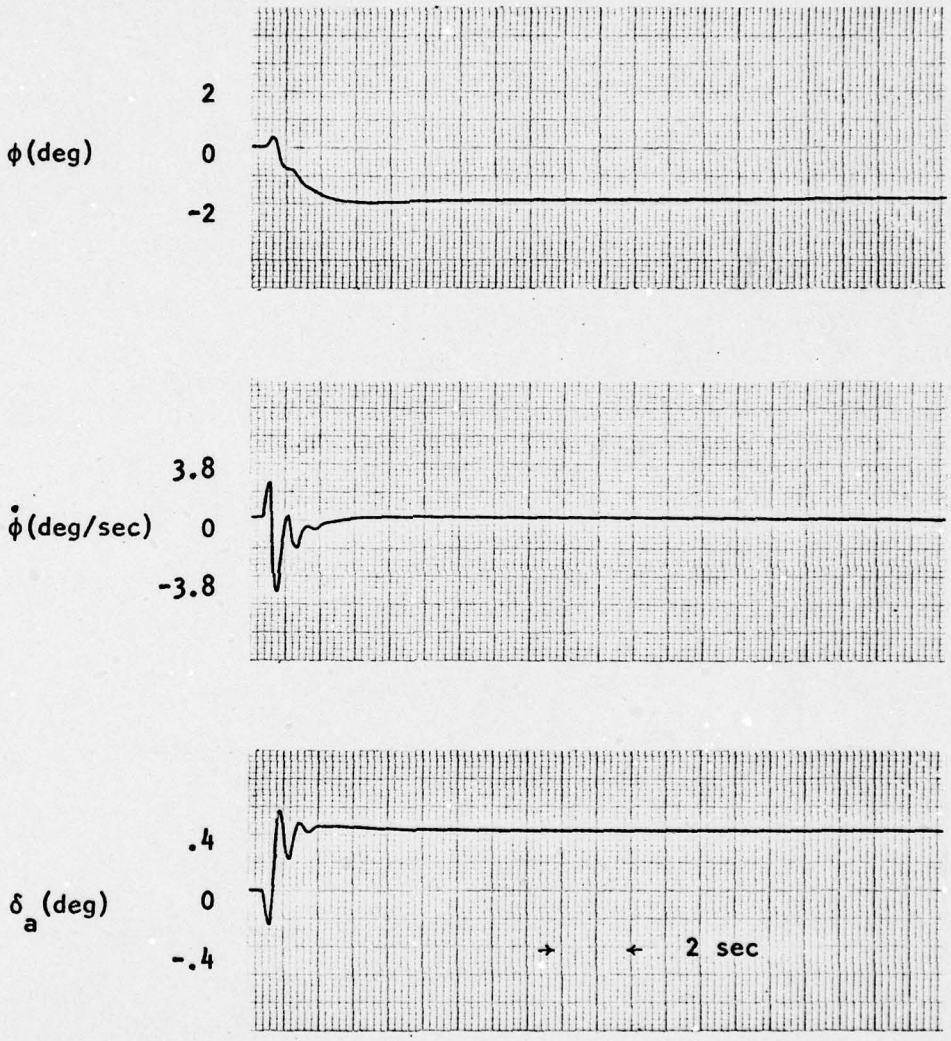


Figure 7-9 : Continued (Analog Yaw Damper Engaged)

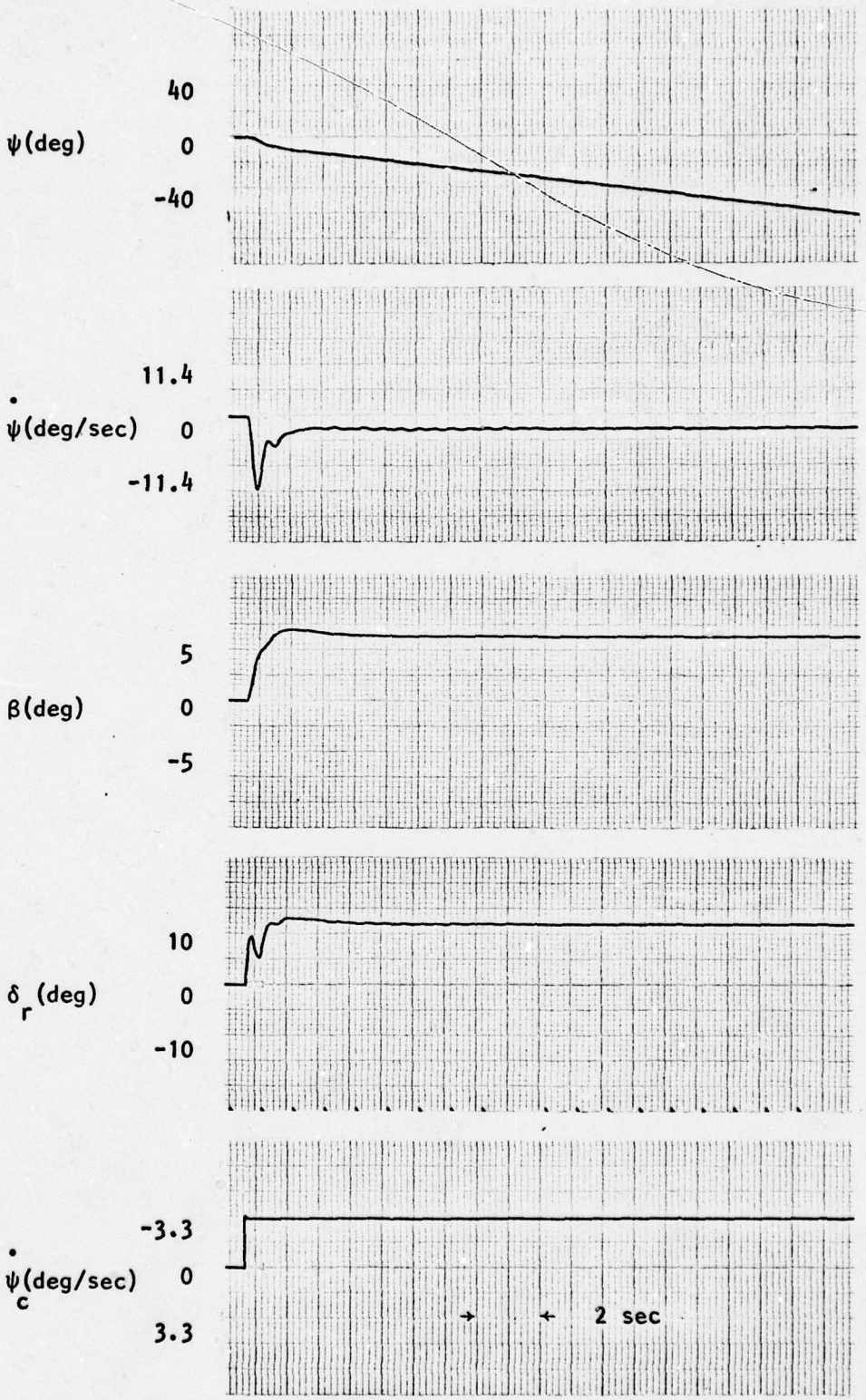


Figure7-10 : RPV Response to 3.3 Deg/Sec Step Yaw Rate Command($\dot{\psi}_c$); Yaw Rate/Accelerometer Loops Closed ; Digital Controller Engaged (Yaw Damper); f_s=50 Hz

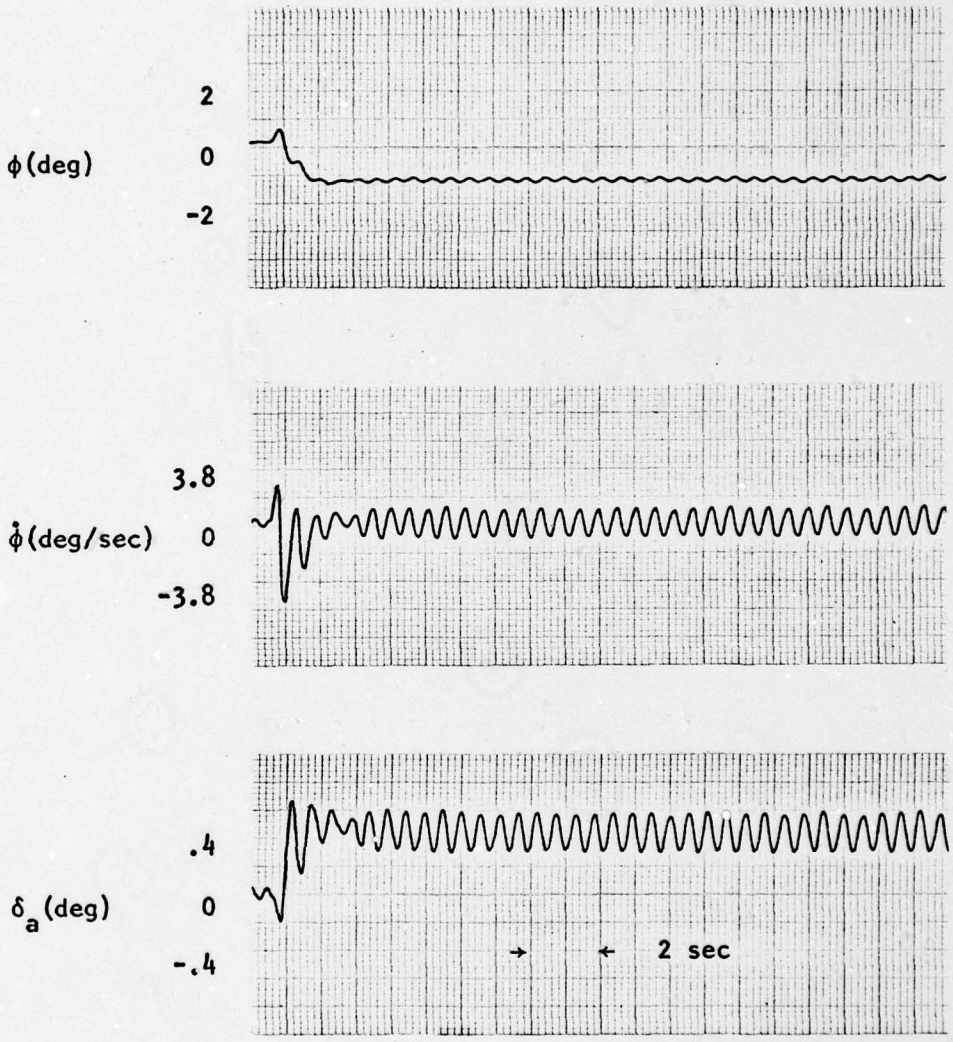


Figure7-10 : Continued (Digital Yaw Damper Engaged; f_s=50 Hz)

In general, digital yaw rate controller performance is, again, quite comparable to the analog controller. No instabilities or potential problem areas exist.

Digital Controller Performance Versus Sampling Frequency Test Results

The next logical course of action was a test of digital controller performance as a function of sampling frequency (f_s). This brief analysis allows the determination of a range of potential sampling frequencies, and insures that the one chosen for final controller implementation will not induce instabilities in the system.

Pitch Attitude Controller. This section presents the four primary pitch attitude controller outputs as a function of f_s. These include ϕ (fig. 7-11), $\dot{\phi}$ (fig. 7-12), α (fig. 7-13) and δ_e (fig. 7-14). Five sampling frequencies are considered: 30 HZ, 10 HZ, 5 HZ, 3 HZ, and 1 HZ. A 20 degree step pitch up forcing function is commanded (θ_c).

Results clearly show that as the f_s decreases to about 2.5 HZ to 3 HZ, successive time response degradation ensues. The controller, however, is still stable but progressively slow and undamped. Below 2.5 HZ the controller appears to go unstable for all four variables. At 1 HZ, the elevators (δ_e) are shown to be oscillating and control is destroyed. Based upon this graphic analysis, it is not inconceivable that the f_s can be decreased to as low as 5 HZ. Other digital controllers referenced in the literature have used sampling frequencies as low as 10 HZ.

Roll Attitude Controller. A similar analysis is made for the roll rate controller. Outputs include ϕ (fig. 7-15), $\dot{\phi}$ (fig. 7-16), β (fig. 7-17), and δ_a (fig. 7-18) for the following f_s: 30 HZ, 10 HZ, 5 HZ. A brief inspection, again, reveals progressive performance degradation as

f_s decreases. The controller goes unstable at about 1.5 HZ (not shown). Roll attitude (ϕ) and sideslip (β) are quite stable for all three cases. Rapid undamping and attendant steady-state oscillations appear for roll rate (ϕ) and aileron deflection (δ_a) as f_s decreases. Because the system is still stable above 1.5 HZ, these tradeoffs may have to be acceptable if f_s is required to decrease from its present value, f_s = 50 HZ.

Yaw Rate Controller. A test of the yaw rate controller was made for several f_s, but not recorded. Results were in agreement with the two controllers above.

Overall Evaluation

This chapter has succeeded in establishing basic digital A/P testing procedures and the results of comparative simulation time response testing. Digital A/P performance has been shown to be very similar to that of the analog A/P. Hence, a major requirement of this study has been satisfied with assurance. Differences in system performance, not unexpectedly, can be attributed to such parameters as finite wordlength, noise, limit cycling, etc.

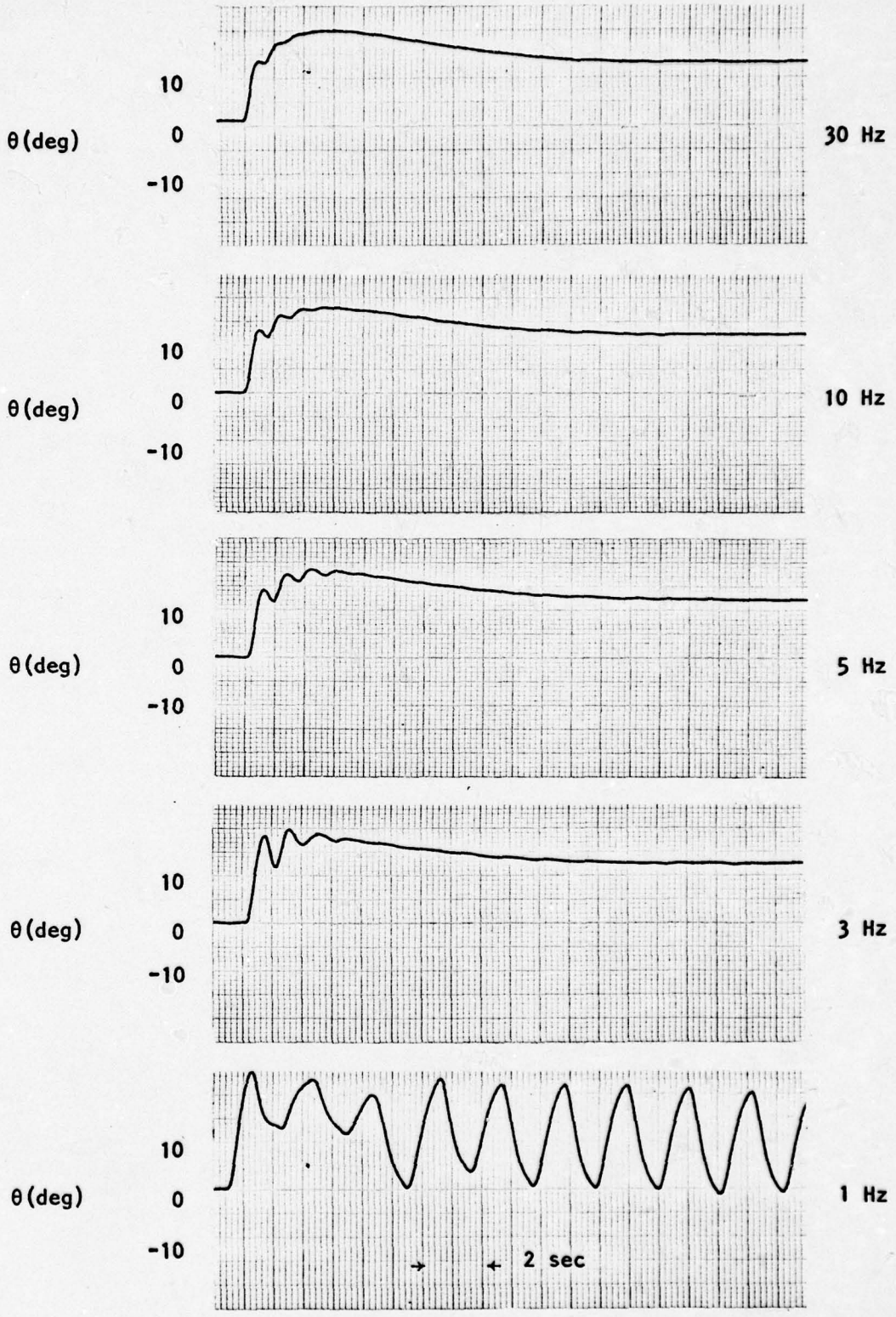


Figure 7-11: Pitch Attitude(θ) Response to 20 Degree Step Pitch Up Command(θ_c) Versus Sampling Frequency(f_s); Full Digital Controller Engaged

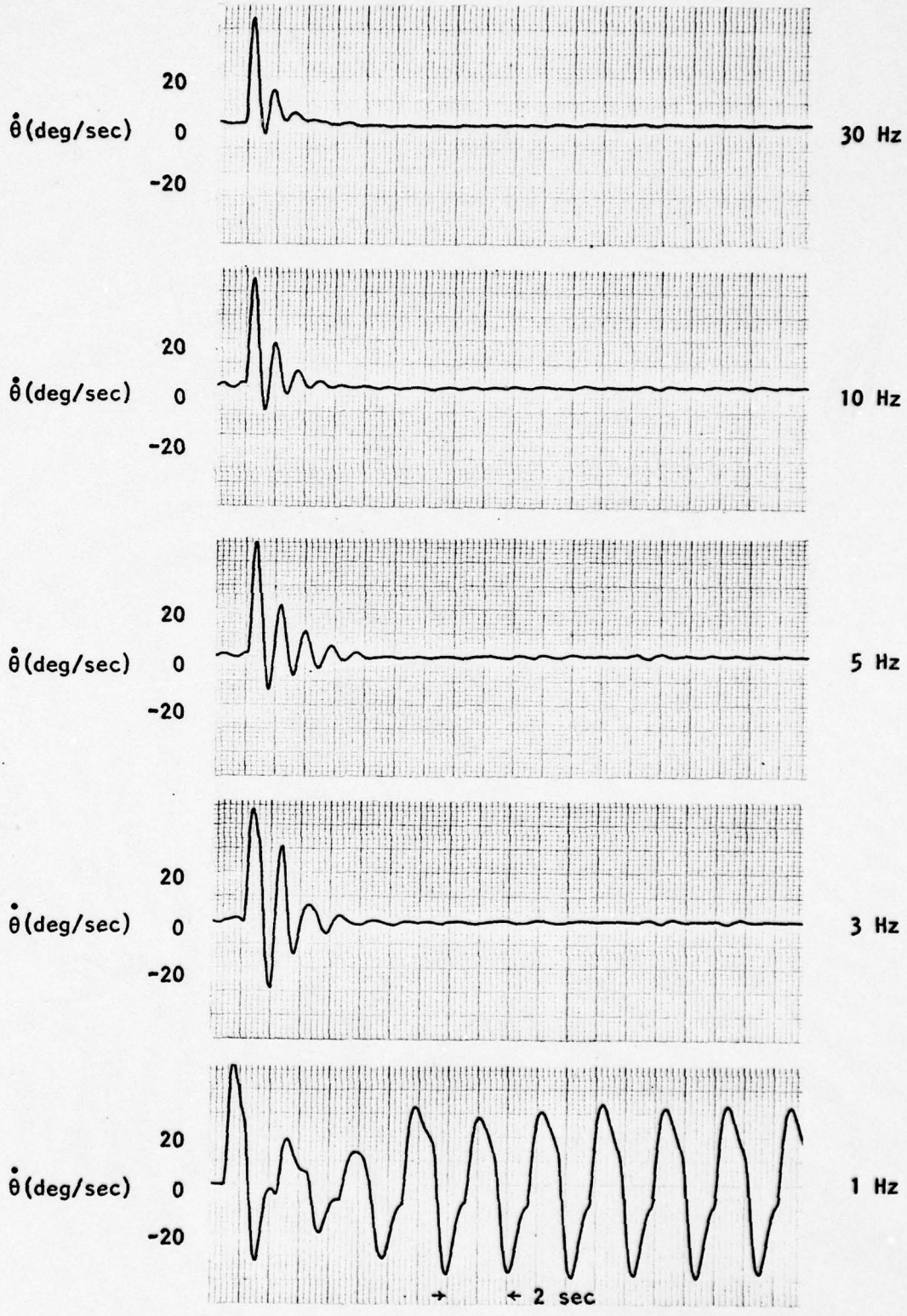


Figure 7-12: Pitch Rate($\dot{\theta}$) Response to 20 Degree Step Pitch Up Command(θ_c) Versus Sampling Frequency(f_s); Full Digital Controller Engaged

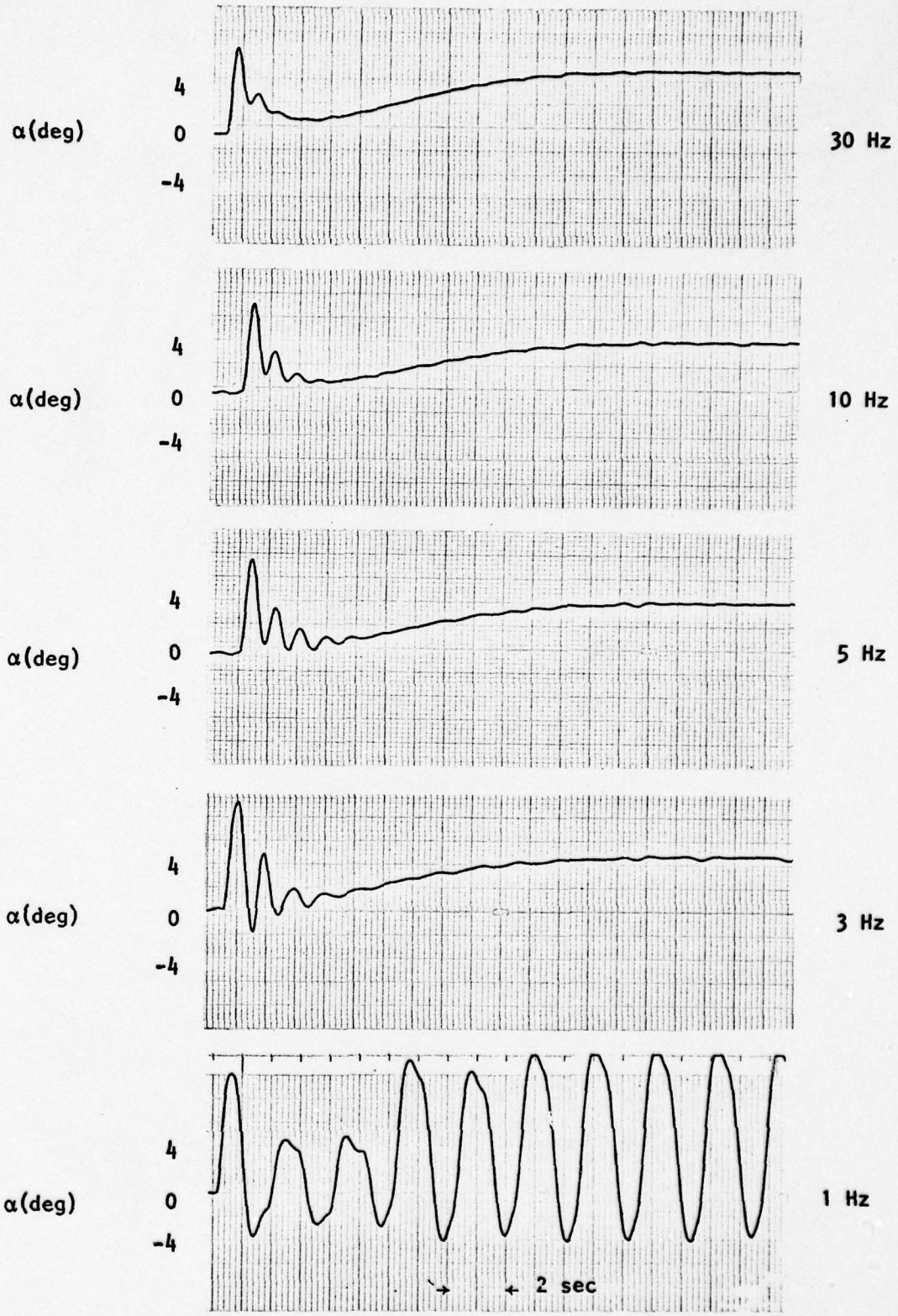


Figure 7-13 : Angle of Attack(α) Response to 20 Degree Step Pitch Up Command(θ_c) Versus Sampling Frequency(f_s); Full Digital Controller Engaged

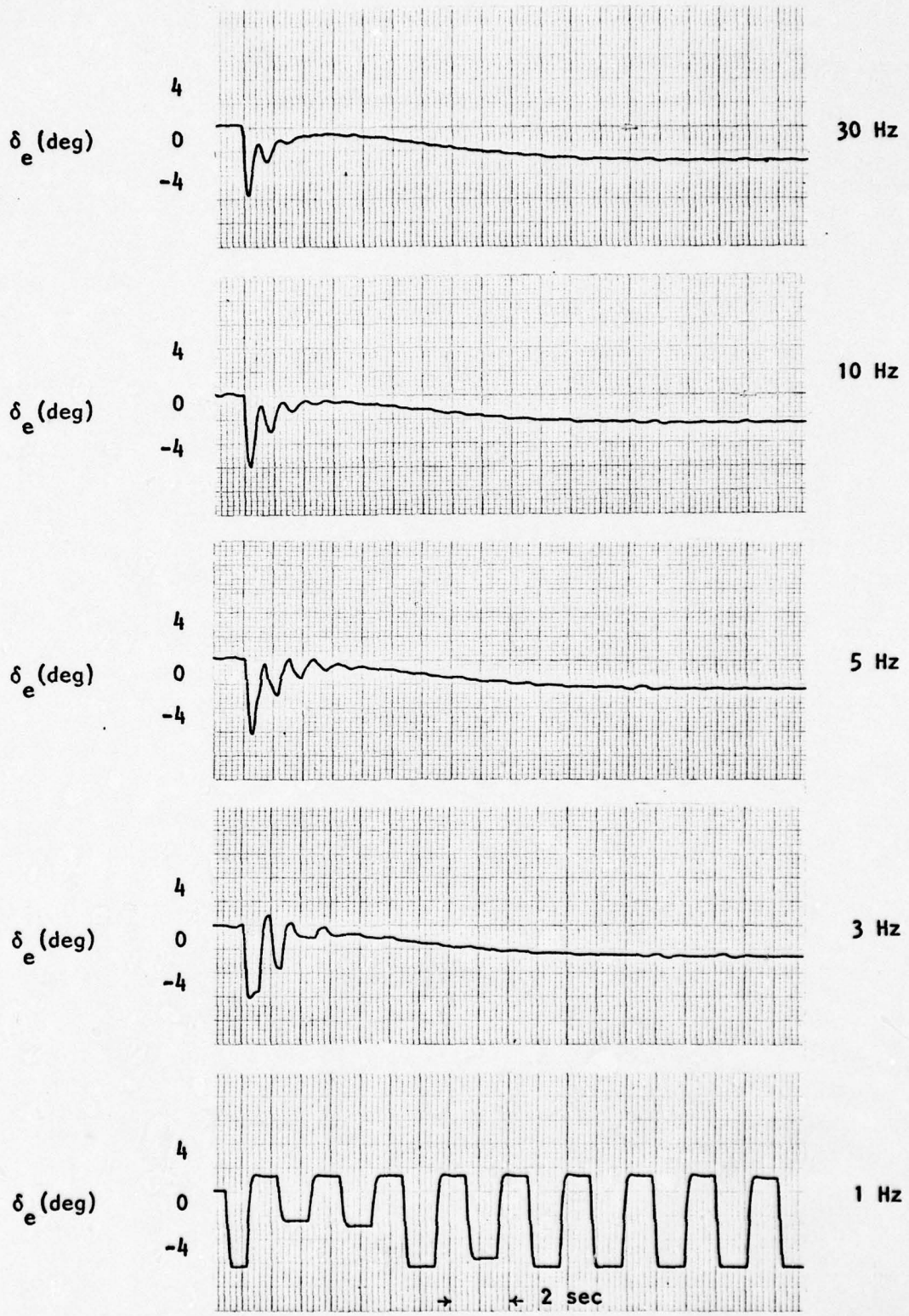


Figure 7-14: Elevator Deflection(δ_e) Response to 20 Degree Step Pitch Up Command(θ_c) Versus Sampling Frequency(f_s); Full Digital Controller Engaged

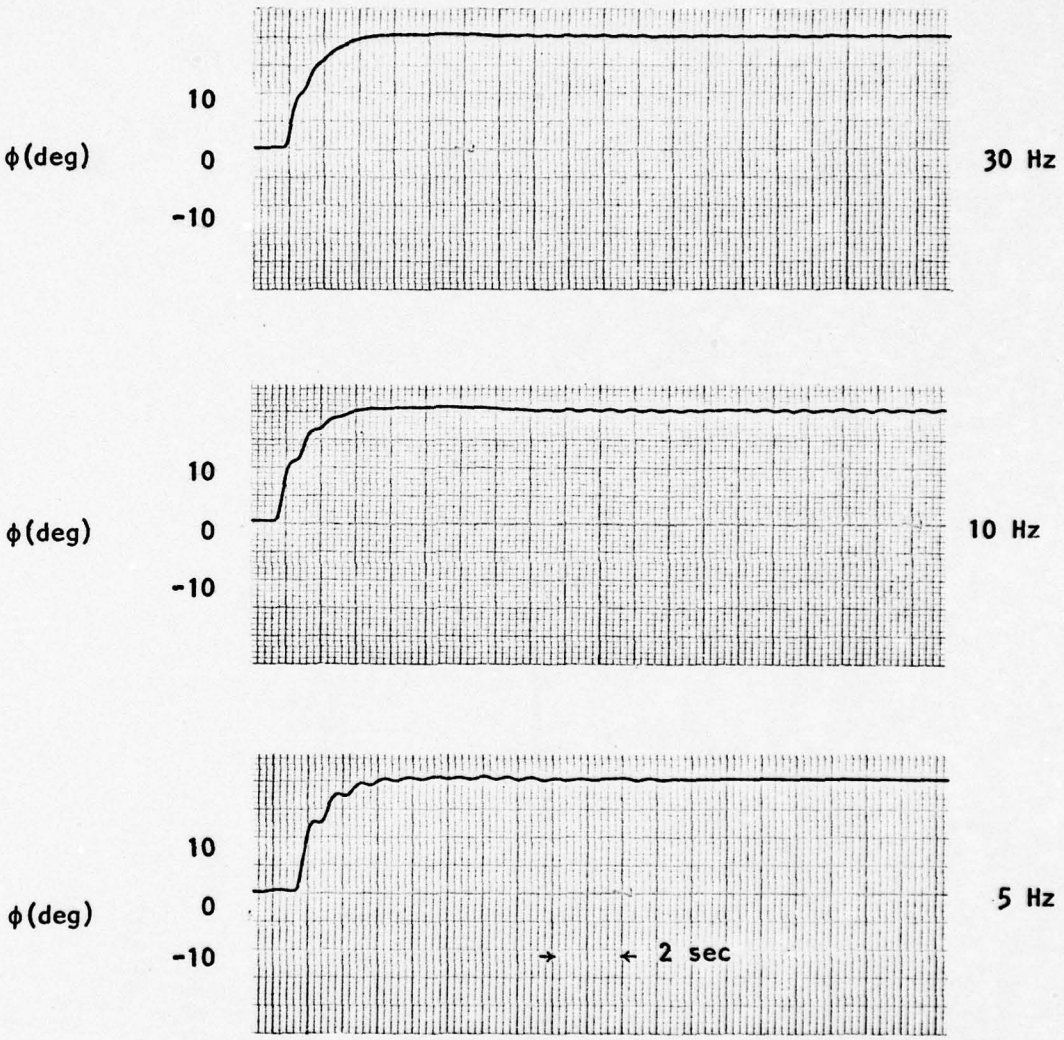


Figure 7-15 : Roll Attitude(ϕ) Response to 20 Degree Step Roll Right Command(ϕ_c) Versus Sampling Frequency(f_s); Full Digital Controller Engaged

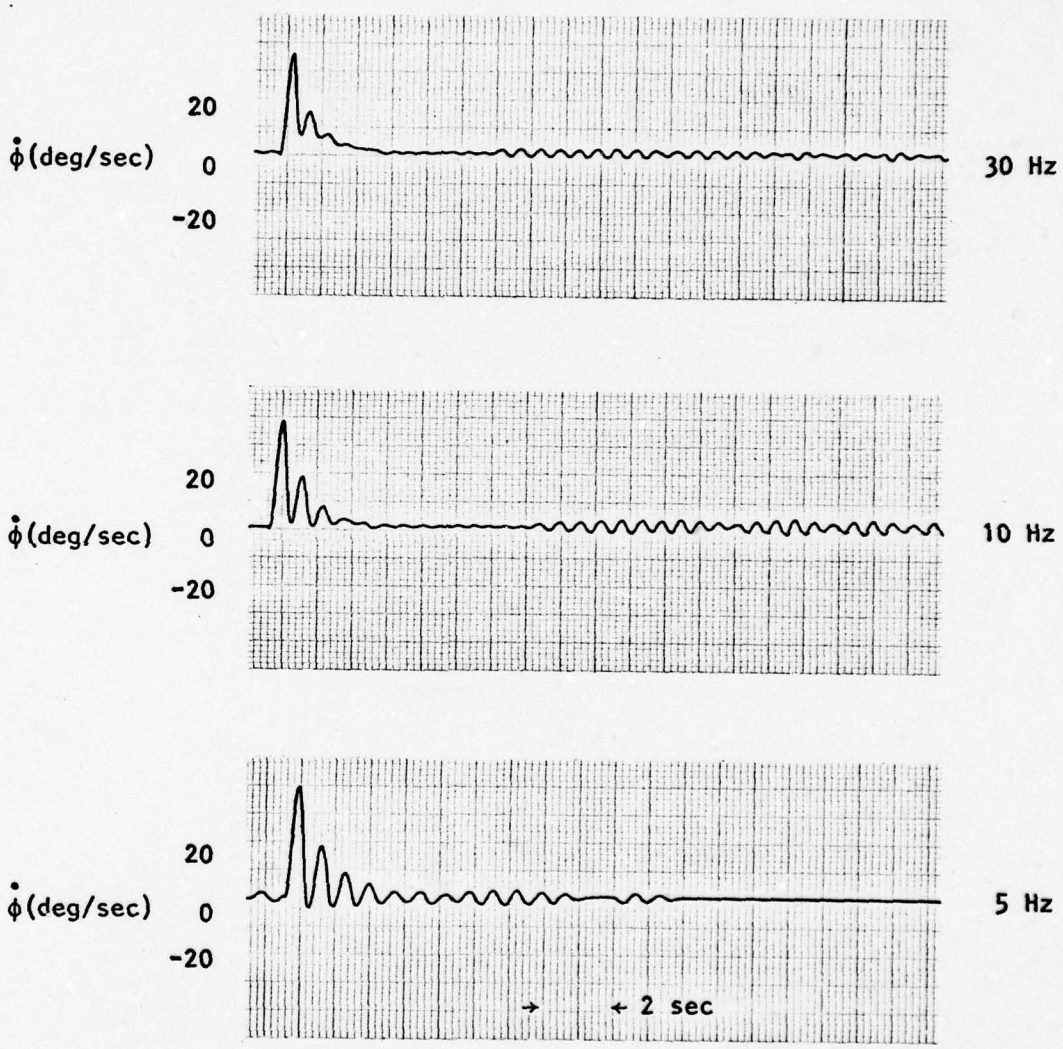


Figure 7-16: Roll Rate($\dot{\phi}$) Response to 20 Degree Step Roll Right Command(ϕ_c) Versus Sampling Frequency(f_s); Full Digital Controller Engaged

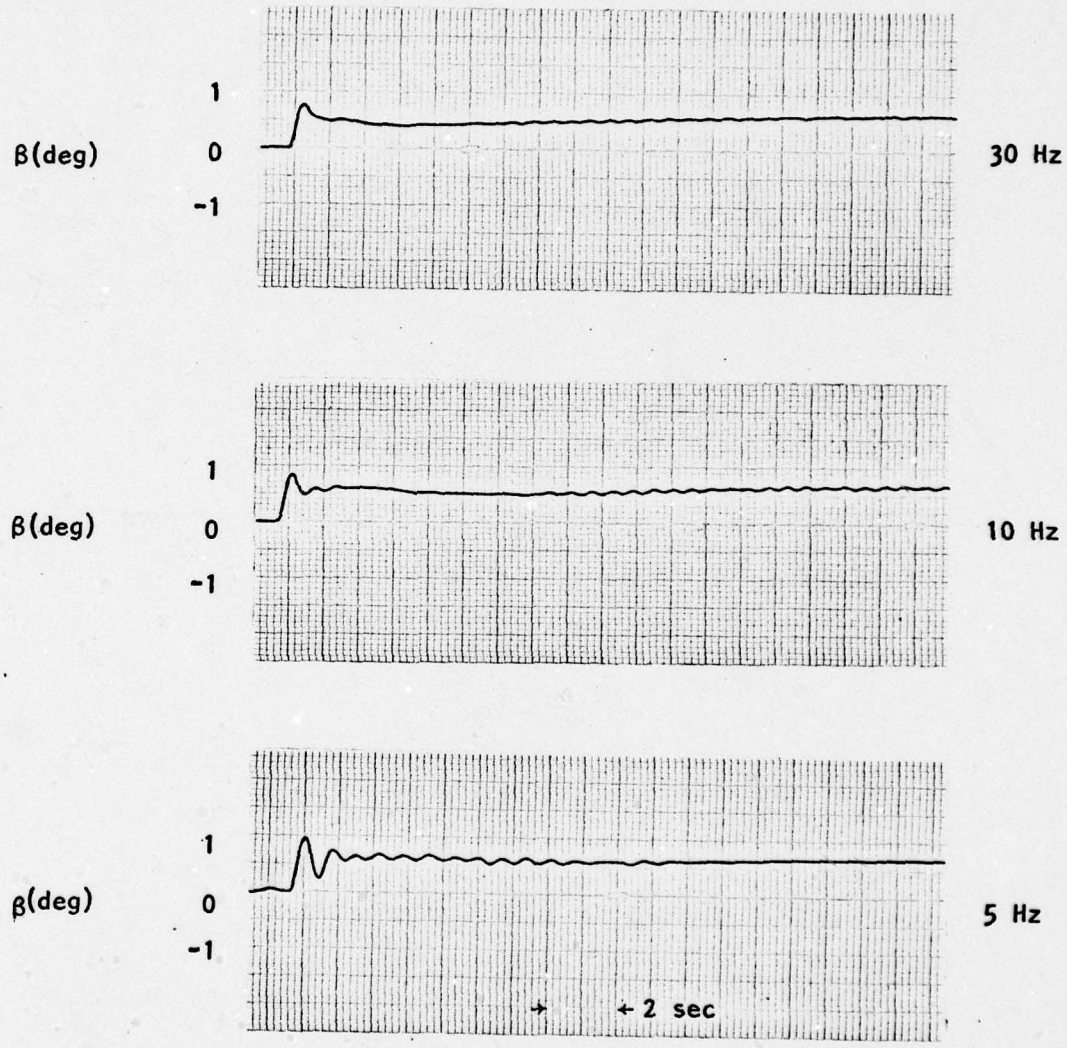


Figure 7-17: Sideslip Angle(β) Response to 20 Degree Step Roll Right Command(ϕ_c) Versus Sampling Frequency(f_s); Full Digital Controller Engaged

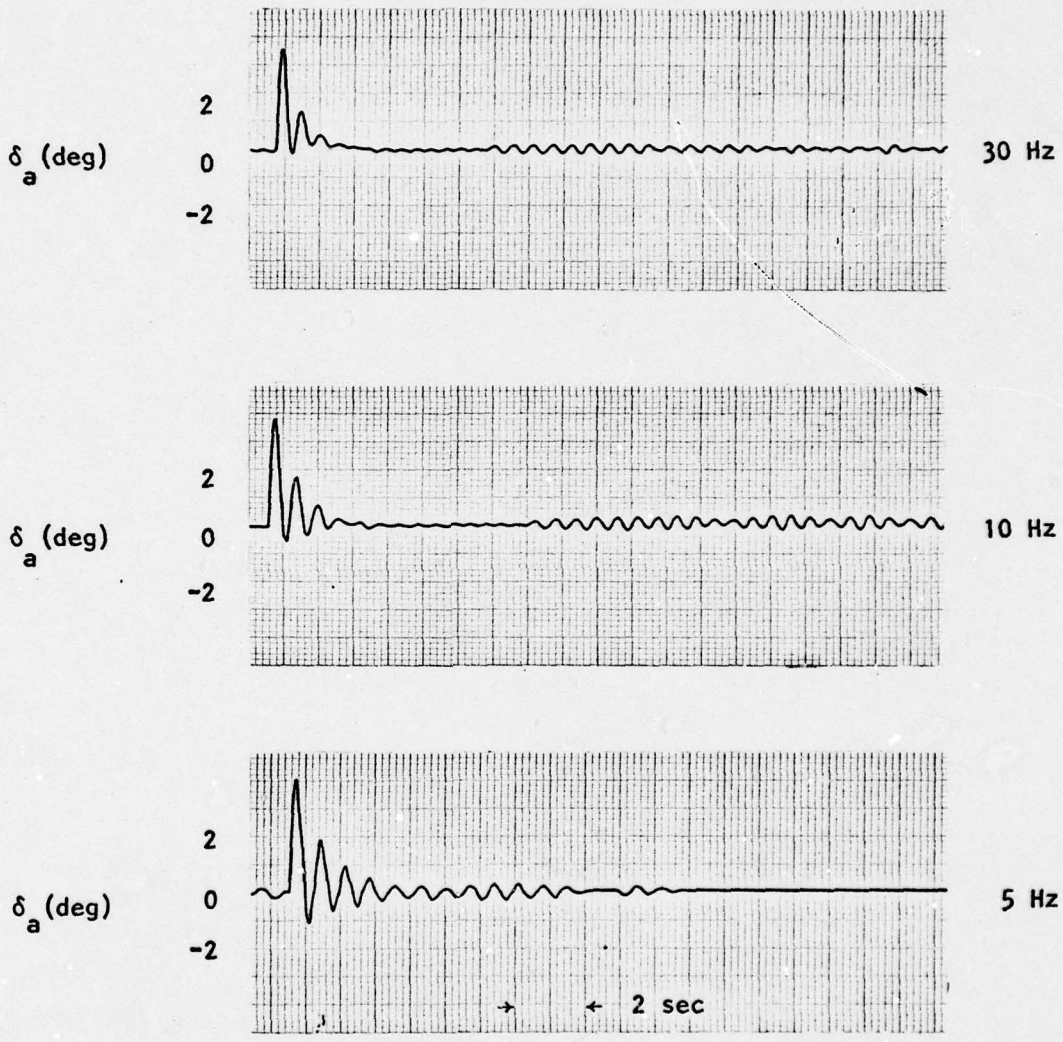


Figure 7-18 : Aileron Deflection(δ_a) Response to 20 Degree Step Roll Right Command(ϕ) Versus Sampling Frequency (f_s); Full Digital Controller Engaged

VIII. Conclusions and Recommendations

A low-cost microprocessor-based digital A/P for the XBQM-106 Mini-RPV has been constructed in breadboard form, and its performance shown to be comparable to the existing analog A/P. Although no cost analysis has been performed, the hardware costs are quite modest since all components are quite inexpensive (and should become even more so in the future). Thus, it appears that a low-cost microprocessor-based A/P is feasible.

As a by-product, three unanticipated beneficial results were realized. First, several errors in the hybrid simulation were corrected when its response failed to agree with the digital A/P. Second, a washout (high-pass) filter in the pitch axis of the existing analog A/P was removed when control analysis showed its effects to be detrimental to system performance. Third, the pendulum sensor in the yaw axis was removed when simulation results showed it to be of questionable value.

This investigation also revealed several additional items that might prove beneficial to the RPV program. First, there are no performance specifications existing within the program. Although probably not needed in the earlier stages of the program, the more sophisticated nature of future developments would surely benefit by a formal definition of system requirements. Second, the servo actuators are the primary limitation to improved control system performance. Although their ultra-low cost makes them an excellent choice for the system as it is used now, any required improvement in control system performance will have to investigate methods of improving actuator characteristics. These might include replace-

ment, redesign, addition of compensators, etc.

Since this project is basically a feasibility study, additional work needs to be completed before a practical digital A/P is realized. The following items have been identified as areas which need additional consideration, and as such, would be suitable for follow-on study:

---Improve performance of the present design by operating the A/D converter and microprocessor at maximum speeds

---Construct a flight testable version of the present breadboard digital A/P

---Incorporate additional functions (such as altitude hold and heading holding) into the digital A/P as they are incorporated into the analog A/P

---Incorporate additional functions into the digital A/P that are not included in the analog A/P (such as failsafe, mixer, etc.)

---Investigate algorithm synthesis for pre-programmed mission profiles (such as loiter patterns)

---Investigate the interfacing of seeker guidance control loops into the A/P

---Improve processor capability by replacing the CDP-1802 microprocessor with the new upward compatible CDP-1804 microprocessor

---Monitor sixteen-bit microprocessor developments (particularly cost and peripheral chip costs) to determine if a low-cost digital A/P can be built using this technology. Obviously, this is useful only if capabilities can be obtained that are not realizable with an eight-bit processor

---Perform a sampled-data control system analysis.

In conclusion, the twin objectives of this study have been met with success. First, a linear control model for the analog A/P has been developed. Thus, future RPV performance criteria can be investigated mathematically rather than by present "hit or miss" methods. Secondly, a

low-cost microprocessor-based A/P has been shown to be feasible, and has the potential for superior performance when compared to the analog A/P if all RPV requirements are considered.

Bibliography

1. -----. "ADC0816/ADC0817 Single Chip Data Acquisition System." A/D Product Specification Sheet. Santa Clara, California: National Semiconductor, September, 1977.

2. Asquith, Frank C. T-6 Digital Autopilot Data Processing Analysis and Specification. Technical Report RG-75-36. Redstone Arsenal, Alabama: 5 March 1975. (AD A010353).

3. Bender, M. A., et al. Digital Flight Control Systems for Tactical Fighters - Vol. III. AFFDL-TR-73-119. Minneapolis, Minnesota: Honeywell, Inc., June 1974. (AD A002687).

4. Blakelock, John H. Automatic Control of Aircraft and Missiles. New York: John Wiley and Sons, Inc., 1965.

5. Brubaker, Thomas A. Digital Filtering Design and Implementation Methods. AFAL-TR-75-211. Department of Electrical Engineering, Colorado State University, Ft. Collins, Colorado: July 1977. (AD A042848).

6. Bucsis, Gary W. Northrop Information Bulletin for MQM-74C Drone Microprocessor-Based Digital Autopilot (Proprietary Information). Newbury Park, California: Northrop Corp., Ventura Division, 15 January 1978.

7. Bursky, Dave. "Microprocessor Selection Guide," Electronic Design, Vol. 25 (No. 21): 55-67 (Oct 11, 1977).

8. Buurma, Jake. "One-Chip System Cuts Cost of Data Acquisition," Electronics, Vol. 51 (No. 9): 131-135 (April 27, 1978).

9. Cadzow, James A. and H. R. Martens. Discrete-Time and Computer Control Systems. New Jersey: Prentice-Hall, Inc., 1970.

10. Callen, T.R. Design and Analysis of an Air-to-Air Missile Using Digital Control. Thesis. Wright-Patterson AFB, Ohio: Air Force Institute of Technology, December 1975. (AD 019853).

11. -----. User Manual for the CDP1802 COSMAC Microprocessor. Somerville, New Jersey: RCA Solid State Division, 1976.

12. -----. CMOS Integrated Circuits. Santa Clara, California: National Semiconductor Corp., 1975.

13. ---.---. Conversion Products Catalog. ADC/DAC Converter Product Guide. Norwood, Massachusetts: Analog Devices, Inc., 1977.

14. -----. D/A Converter Product Guide. NE5018. Sunnyvale, California: Signetics Corporation, August 1977.

15. D'Azzo, John J. and Constantine H. Houpis. Linear Control and System Analysis and Design. New York: McGraw-Hill, Inc., 1975.

16. -----. "DC/DC Converter". Burr-Brown Product Guide. Tucson, Arizona: Burr-Brown, 1978.

17. Derrick, Milford R. Controllers and Autopilot Modes for RPV Simulation Program. AMRL-TR-74-90. McLean, Virginia: Adaptronics, Inc., April 1976. (AD A025792).

18. Distephano, Joseph J., III, et al. Feedback and Control Systems. Schaum's Outline Series. New York: McGraw-Hill Book Company, 1967.

19. -----. "Elf II". Elf II Microprocessor. New Milford, Connecticut: Netronics R & D Limited, 1977.

20. Emmert, Robert I., et al. Digital Control Techniques for Terminally Guided Air-to-Surface Weapons. AFATL-TR-73-74. Eglin AFB, Florida: Rockwell International Corporation, March 1973. (AD 912285L).

21. Enke, Stephen E. Direct Digital Design of the Flight Controller for the Pitch Axis of the A-7D Aircraft. Thesis. Wright-Patterson AFB, Ohio: Air Force Institute of Technology, December 1977.

22. Garner, Douglas H. and Harold E. Poole. Development and Flight Tests of a Gyro-less Wing Leveler and Directional Autopilot. Hampton, Virginia: Langley Research Center, April 1974.

23. -----. "Giant Board". Elf II Microprocessor Operating System. New Milford, Connecticut: Netronics R & D Limited, 1977.

24. -----. Engineering Product Handbook. ADC/DAC Converter Product Guide. Santa Ana, California: Datel Systems, Inc., 1977.

25. Hilburn, John L. and Johnson, David E. Manual of Active Filter Design. New York: McGraw-Hill Book Company, 1973.

26. Kryske, L. E. and E. S. Sutherland. Advanced Digital Technology for Missiles--Phase III. Culver City, California: Hughes Aircraft Company, February, 1974. (AD 921368).

27. Kuo, Benjamin C. Digital Control Systems. Illinois: SRL Publishing Company, 1977.

28. Larimer, Stanley. TOTAL. Unpublished User's Guide to CAD Computer Program. Wright-Patterson AFB, Ohio: Air Force Institute of Technology, 1977.

29. Linder, Larry J. and James Bindner. XBQM-106 Autopilot Development Status Report--Yuma Proving Ground, Ariz. AFFDL-TM-78-63-FXG. Wright-Patterson AFB, Ohio: KBG Corporation, February-March 1978.

30. McGlynn, Henry J. Computer Programs for Computational Assistance in the Design of Aircraft Control Systems. ASD-TR-72-112. Wright-Patterson AFB, Ohio: Aeronautical Systems Division, December 1972. (AD 758781).

31. McGlynn, Henry J., Jr. TTYLON/TTYLAT User Guide: A Teletype-Interactive Longitudinal/Lateral Aircraft Transfer Function Program. Unpublished Report. Wright-Patterson AFB, Ohio: Air Force Flight Dynamics Laboratory, 1972.

32. McRuer, Duane and Irving Ashkenas. Aircraft Dynamics and Automatic Control. Princeton, New Jersey: Princeton University Press, 1973.

33. -----. Melpar Digital RPV Autopilot Bulletin (Incomplete). Unpublished Report. Falls Church, Virginia: Melpar Corporation, Division of E-Systems, 1977.

34. -----. Memory Data Book. Santa Clara, California: National Semiconductor Corp., 1976.

35. MPG-180A. COSMAC Microprocessor Product Guide. Somerville, New Jersey: RCA Corporation, 1977.

36. -----. TRW Multiplier-Accumulator Parallel 8-Bit. TDC1008J High Speed Multiplier Chip Specification Sheet. Redondo Beach, California: TRW LSI Products, 1978.

37. Oppenheim, Alan V., et al. Digital Signal Processing. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1975.

38. -----. "Precision Pendulum (CP17-0601-3)." Product Specification Sheet. San Diego, California: Humphrey, Inc., (Undated).

39. Pietrzak, Paul E. and John T. Cors. Data Package for Simulation of the XBQM-106 Mini-RPV, Baseline and Sideforce Configurations. Technical Memorandum SD 26 TM-78-01. Wright-Patterson AFB, Ohio: PLS/RPV System Program Office, June 1978.

40. -----. AFFDL Flight Control Simulation Computer Program for the Close Air Support Mission (DAIS). Honeywell User's Manual. Minneapolis, Minnesota: Honeywell Inc., November 1975.

41. Prosser, Charles F., et al. RPV Flying Qualities Design Criteria. AFFDL-TR-76-125. Columbus, Ohio: Rockwell International Corp., Missile System Division, December 1976.

42. Roskam, Jan. Flight Dynamics of Rigid and Elastic Airplanes. Kansas: Roskam Aviation and Engineering Corporation, 1976.

43. Schmidt, Lynn A. "Designing Programmable Digital Filters for LSI Implementation," Hewlett-Packard Journal: 15-23 (September 1978).

44. -----. "Futaba FP-S14 (Large Servo) Handling Instructions." Product Specification Sheet. Compton, California: Futaba Industries U.S.A., (Undated).

45. Smith, Bruce A. "Digital Processor to Add to Drone Capabilities," Aviation Week & Space Technology: 70 (May 22, 1978).

46. -----. Model 262 "Star" Second Quarterly Technical Report. TRA 26263-3. San Diego, California: Teledyne Ryan Aeronautical, 24 October, 1975. (AD B007416).

47. Stengel, Robert F. "Digital Flight Control Research Using Microprocessor Technology." Unpublished Bulletin. Department of Aerospace and Mechanical Sciences. Princeton University. July 1978.

48. Summers, R. R. Application of the Aerospace Multiprocessor to the A-7D Flight Control System. Thesis. Wright-Patterson AFB, Ohio: Air Force Institute of Technology, June 1973.

49. -----. "Linear Integrated Circuits--555 Timer." Timer Specification Sheet. Signetics Corporation. (Undated).

50. -----. The TTL Data Book for Design Engineers. Dallas, Texas: Texas Instruments, Inc., 1976.

51. -----. Vertical Gyro (VG34-0201-1). Product Specification Sheet. San Diego, California: Humphrey, Inc., (Undated).

Appendix A

The purpose of this appendix is to augment the discussion in Chapter II concerning RPV transfer functions. In particular, this section includes the following:

--Transfer function derivation

--Transfer function listings: tables A-12 ~ A-13a,b

--Transfer function frequency responses: fig. A-1a ~ A-11

--Transfer function time responses: fig. A-12a ~ A-27b

--Transfer function root loci: fig. A-28a ~ A-41

--Free RPV airframe responses (hybrid simulation): fig. A-42 ~ A-49

Derivation of Linear XBQM-106 RPV Transfer Functions

A complete control analysis of the longitudinal and lateral-directional axes requires that the aircraft transfer functions be known. An interactive computer program, the AFFDL McGlynn program (1973), is used via INTERCOM, to derive the representative transfer functions for three flight conditions (Ref:31):

	Altitude (ft,msl*)	Weight (lb)	Mach
Case I	1000	135	.125
Case II	3000	135	.125
Case III	6000	135	.125

* Mean Sea Level

Program inputs consist of desired flight case parameters and aerodynamics data. Laplace transfer functions are generated as program output. Aerodynamic data called for in the program is extracted from current wind tunnel data provided by the Mini-RPV Group/AFFDL (Ref:39). "Standard"

control surface deflection notation is used to insure proper sign convention of the input data (see fig. 2-3, Chap. II). Each requested output transfer function (for Case II only) is listed in both standard (root locus) and alternate (Bode) form in Tables A-12 and A-13. The AFFDL McGlynn program consists of two subprograms: TTYLON (longitudinal axis), and TTYLAT (lateral axis).

TTYLON

TTYLON calculates either the full longitudinal axis (phugoid and short-period) transfer functions or the short-period approximation transfer functions. This study deals with only the full transfer functions. Input parameters consist of specified non-dimensional stability derivatives and RPV constants (flight case parameters). The input aero data is the same for all three cases and is listed in Table A-1. Intermediate output data, depicted in Tables A-2, A-3, and A-4 shows the following information:

--Flight case data - echo of input constants

--Non-dimensional derivative matrix - "trim-conditioned" derivatives calculated for Table A-1

--Dimensional derivative parameter matrix - body axes derivatives calculated from non-dimensional derivative matrix

TTYLAT

TTYLAT calculates both the roll and yaw axis transfer functions. Input aero data is the same for all three cases and is listed as a non-dimensional derivative matrix in Tables A-5, A-6, and A-7. These three tables also present the following information:

--Flight case data - echo of input constants

--Non-dimensional derivative matrix - echo of input aero data

--Dimensional derivative parameter matrix - body axes derivatives calculated from non-dimensional derivative matrix

Once the required input data is entered interactively into either program, the desired transfer functions are chosen from a menu of available options. The resulting transfer functions are then calculated in both polynomial and factored form.

Linear Longitudinal Equations

A brief description of the longitudinal transfer function derivation is essential to an understanding of the linearization process involved. McGlynn (Ref:30;31) describes the linear longitudinal equations of motion in body axes as follows:

$$\Delta X + \Delta T \cos \varepsilon_T = m(\dot{u} + W_o q + \theta_g \cos \theta_o)$$

$$\Delta Z - \Delta T \sin \varepsilon_T = m(U_o \dot{\alpha} - U_o q + \theta_g \sin \theta_o)$$

$$\Delta M + \Delta T Z_T \cos \varepsilon_T = I_{yy} \dot{q}$$

$$\dot{\theta} = q$$

Variable and coefficient definitions are listed in (Ref:30). These equations are transformed into a standard perturbation matrix model of the form

$$\overline{A}\,\overline{X} = \overline{B}\,\overline{U}$$

which relates forward velocity (U), angle of attack (α), pitch rate (q),

and pitch attitude (θ) to elevator displacement (δ_e) and thrust (T). The model is expressed in Laplace form by:

$$\underset{\overline{A}}{\begin{bmatrix} (1-X_{\dot{u}})S-X_u & -X_{\dot{\alpha}}S-X_\alpha & W_o-X_q & gCOS\theta_o \\ -Z_{\dot{u}}S-Z_u & (U_o-Z_{\dot{\alpha}})S-Z_\alpha & -U_o-Z_q & gSIN\theta_o \\ -M_{\dot{u}}S-M_u & -M_{\dot{\alpha}}S-M_\alpha & S-M_q & 0 \\ 0 & 0 & -1 & S \end{bmatrix}} \underset{\overline{X}}{\begin{bmatrix} U \\ \alpha \\ q \\ \theta \end{bmatrix}} = \underset{\overline{B}}{\left[\begin{array}{c|c} X_{\delta_e} \frac{COS\varepsilon_T}{m} & \\ Z_{\delta_e} -\frac{SIN\varepsilon_T}{m} & \\ M_{\delta_e} \frac{Z_T COS\varepsilon_T}{I_{yy}} & \\ 0 & 0 \end{array}\right]} \underset{\overline{U}}{\begin{bmatrix} \delta_e \\ \Delta T \end{bmatrix}}$$

The body axes dimensional derivative parameters used in this matrix (eg. X_u, X_x) are derived in the program by manipulation of the equations presented in Tables A-8 and A-9. Conversion equations in Table A-9 require a program calculated trim angle of attack which is not supplied as an input parameter. Solution of the perturbation matrix for specific transfer functions (eg. θ/δ_e, θ/δ_e) follows the conventional Cramer's Rule technique (Ref:4;17;21;30;31;32;42).

Linear Lateral Equations

From (Ref:30;31), the set of linear lateral equations of motion (body axes) is described by:

$$\Delta Y = m(V_T\dot{\beta} - W_o p + U_o r - \phi g COS\theta_o)$$

$$\Delta L = I_{xx}\dot{p} - I_{xz}\dot{r}$$

$$\vdots$$

$$\dot{\phi} = p + r\tan\theta_o$$

$$\dot{\psi} = r/COS\theta_o$$

These differential equations are transformed into the Laplace perturbation matrix model which consists of the following:

$$\underset{\bar{A}}{\begin{bmatrix} (V_T - Y_{\dot{\beta}})S - Y_\beta & -W_o - Y_p & U_o - Y_r & -g\cos\theta_o \\ -L_{\dot{\beta}}S - L_\beta & S - L_p & -\left(\frac{I_{yz}}{I_{xx}}\right)S - L_r & 0 \\ -N_{\dot{\beta}}S - N_\beta & -\left(\frac{I_{xz}}{I_{zz}}\right)S - N_p & S - N_r & 0 \\ 0 & -1 & -\tan\theta_o & S \end{bmatrix}} \underset{\bar{X}}{\begin{bmatrix} \beta \\ P \\ r \\ \phi \end{bmatrix}} = \underset{\bar{B}}{\left[\begin{array}{c:c} Y_{\delta_A} & Y_{\delta_R} \\ \hdashline L_{\delta_A} & L_{\delta_R} \\ \hdashline N_{\delta_A} & N_{\delta_R} \\ \hdashline 0 & 0 \end{array}\right]} \underset{\bar{U}}{\begin{bmatrix} \delta_A \\ \delta_R \end{bmatrix}}$$

The body axes dimensional derivative parameters (eg. Y_β, L_p) are derived in the program by manipulations presented in Tables A-10 and A-11. Matrix solution for desired transfer functions is accomplished, again, via Cramer's Rule (Ref:4;17;21;30;31;32;42).

Transfer Functions

Table A-12 includes longitudinal transfer functions for pitch attitude (θ), pitch rate ($\dot{\theta}$), angle of attack (α) and forward velocity (U, rad/sec) related to elevator deflection (δ_e). It is important to note that both the phugoid and short-period modes are incorporated in the equations (Ref 4:35).

Table A-13 includes the more common lateral-directional (roll and yaw) transfer functions. Roll (ϕ), roll rate ($\dot{\phi}$), yaw (ψ), yaw rate ($\dot{\psi}$), and sideslip (β) are expressed relative to both aileron deflection (δ_a) and rudder deflection (δ_r). Roll subsidence, spiral divergence, and dutch roll factors are implicit in the functions (Ref 4:118). Three flight cases are described.

Frequency Responses

The following transfer function frequency responses are depicted for each axis:

Pitch : fig. A-1a ~ A-3b

Roll : fig. A-4a ~ A-7

Yaw : fig. A-8a ~ A-11

Note that frequency responses are also shown with the actuator servo multiplied by the basic transfer function. These can be used directly to determine system gain and bandwidth in the forward loop.

Time Responses

The following transfer function time responses to selected forcing functions are shown for each axis:

Pitch : fig. A-12a ~ A-16b

Roll : fig. A-17a ~ A-20

Yaw : fig. A-21a ~ A-27b

Again, actuator responses are included.

Root Loci

The following transfer function root loci are depicted for each axis:

Pitch : fig. A-28a ~ A-33b

Roll : fig. A-34a ~ A-37

Yaw : fig. A-38a ~ A-41

Note that magnified loci are shown for selected functions so as to aid in analysis. Actuator inclusions are especially helpful in showing loci branching effects.

Free RPV Airframe Responses

The following time responses are shown for similar forcing functions used in fig. A-12a ~ A-27b:

Pitch	:	fig. A-42 ~ A-43
Roll	:	fig. A-44 ~ A-45
Yaw	:	fig. A-46 ~ A-49

These strip chart recordings were generated from a six degree-of-freedom (6DOF) non-linear RPV model on hybrid simulation.

Table A-1

Longitudinal Axis Input Aero Data (Cases I, II, III)

Mnemonic*	Value	Units
MACH	0.125	----
CL (AOA=0)	0.340	----
CLA	0.094	per deg
CLDE	0.012	per deg
CLQ	7.0	per rad
CLAD	1.15	per rad
CLMAX	1.2	----
CM (AOA=0)	-0.06	----
CMA	-0.03	per deg
CMDE (AOA)	-0.043	per deg
CMQ	-17.	per rad
CMAD	-4.	per rad
CDMIN	0.048	----
CL (CDMIN)	0.18	----
KCL2	0.0565	----
DEMIN	0.	deg
KDE2	0.	per deg^2
TMAX	50.	lbs

:

*Refer to list of symbols and definitions

Table A-2

Case I (Longitudinal Axis)

Flight Case Data

MACH	.12500	VFPS*	139.30	VSND*	1114.37
ALT	1000.	QDYN*	22.39	DEN*	.0023080
WGHT	135.	IYY	25.	AREA	18.18
CG	5.420	REFCG	5.420	CHORD	1.65
GAMMA*	0.000	EPSTH	0.000	ZT	-.798
AOA*	.165	ELEV*	-2.002	THRST*	20.07

Non-Dimensional Derivative Matrix

Perturbation	CD	CL	CM
O (per rad)	.04929653	.33148420	.02115278
M (per rad)	0.00000000	0.00000000	0.00000000
A (per deg)	.00160907	.09400000	-.03000000
AD (per rad)	0.00000000	1.15000000	-4.00000000
Q (per rad)	0.00000000	7.00000000	-17.00000000
DE (per deg)	.00020541	.01200000	-.04300000

Dimensional Derivative Parameter Matrix*

Perturbation	X	Z	M
U (per rad)	-.67850E-01	-.45105E+00	.89411E-02
UD (per rad)	-.39358E-07	.13664E-04	.12911E-04
A (per rad)	.24707E+02	-.52742E+03	-.45298E+02
AD (per rad)	.19033E-02	-.66079E+00	-.62437E+00
Q (per rad)	.11586E-01	-.40222E+01	-.26536E+01
DE (per rad)	-.94970E+00	-.66708E+02	-.64932E+02

* Internally Calculated

Table A-3

Case II (Longitudinal Axis)

Flight Case Data

MACH	.12500	VFPS*	138.33	VSND*	1106.63
ALT	3000.	QDYN*	20.81	DEN*	.0021749
WGHT	135.	IYY	25.	AREA	18.18
CG	5.420	REFCG	5.420	CHORD	1.65
GAMMA*	0.000	EPSTH	0.000	ZT	-.708
AOA*	.457	ELEV*	-2.211	THRST*	18.82

Non-Dimensional Derivative Matrix

Perturbation	CD	CL	CM
O (per rad)	.04975951	.35647010	.02135203
M (per rad)	0.00000000	0.00000000	0.00000000
A (per deg)	.00187447	.09400000	-.03000000
AD (per rad)	0.00000000	1.15000000	-4.00000000
Q (per rad)	0.00000000	7.00000000	-17.00000000
DE (per deg)	.00023929	.01200000	-.04300000

Dimensional Derivative Matrix*

Perturbation	X	Z	M
U (per rad)	-.62670E-01	-.43688E+00	.99904E-02
UD (per rad)	-.28498E-06	.35691E-04	.33724E-04
A (per rad)	.26298E+02	-.49034E+03	-.42086E+02
AD (per rad)	.49369E-02	-.61830E+00	-.58423E+00
Q (per rad)	.30053E-01	-.37638E+01	-.24831E+01
DE (per rad)	-.74112E+00	-.61995E+02	-.60339E+02

*Internally Calculated

Table A-4

Case III (Longitudinal Axis)

Flight Case Data

MACH	•	.12500	VFPS*	•	136.86	VSND*	•	1094.91
ALT	•	6000.	QDYN*	•	18.60	DEN*	•	.0019863
WGHT	•	135.	IYY	•	25.	AREA	•	18.18
CG	•	5.420	REFCG	•	5.420	CHORD	•	1.65
GAMMA*	•	0.000	EPSTH	•	0.000	ZT	•	-.708
AOA*	•	.948	ELEV*	•	-2.562	THRST*	•	17.15

Non-Dimensional Derivative Matrix

Perturbation	CD	CL	CM
0 (per rad)	.05069293	.39831758	.02175485
M (per rad)	0.00000000	0.00000000	0.00000000
A (per deg)	.00231897	.09400000	-.03000000
AD (per rad)	0.00000000	1.15000000	-4.00000000
Q (per rad)	0.00000000	7.00000000	-17.00000000
DE (per deg)	.00029604	.01200000	-.04300000

Dimensional Derivative Parameter Matrix*

Perturbation	X	Z	M
U (per rad)	-.55397E-01	-.41714E+00	.11507E-01
UD (per rad)	-.11164E-05	.67502E-04	.63789E-04
A (per rad)	.28515E+02	-.43874E+03	-.37611E+02
AD (per rad)	.92373E-02	-.55853E+00	-.52781E+00
Q (per rad)	.56242E-01	-.34007E+01	-.22438E+01
DE (per rad)	-.45057E+00	-.55435E+02	-.53947E+02

* Internally Calculated

Table A-5

Case I (Lateral Axis)

Flight Case Data

MACH	•	.12500	ALT	•	1000.	WGHT	• 135.
AOA	•	0.00	SPAN	•	11.17	AREA	• 18.18
IXX	•	9.	IZZ	•	32.	IXZ	• -2.
QDYN*	•	22.39	VFPS*	•	139.30	DEN*	• .0023080

Non-Dimensional Derivative Matrix

Perturbation	CY	CL	CN
B (per deg)	-.01800000	-.00060000	.00290000
P (per rad)	0.00000000	-.45600000	-.02600000
R (per rad)	.75000000	.07500000	-.35000000
BD (per rad)	0.00000000	0.00000000	0.00000000
DA (per deg)	.00180000	.00420000	-.00021000
DR (per deg)	.00265000	.00015000	-.00135000

Dimensional Derivative Parameter Matrix*

Perturbation	Y	L	N
B (per rad)	-.10006E+03	-.17113E+02	.23291E+02
P (per rad)	0.	-.90989E+01	-.11238E+00
R (per rad)	.29167E+01	.14965E+01	-.19666E+01
BD (per rad)	0.	0.	0.
DA (per rad)	.10006E+02	.11979E+03	-.16866E+01
DR (per rad)	.14731E+02	.42783E+01	-.10843E+02

* Internally Calculated

Table A-6

Case II (Lateral Axis)

MACH	.12500	ALT	3000.	WGHT	135.
AOA	0.00	SPAN	11.17	AREA	18.18
IXX	9.	IZZ	32.	IXZ	-2.
QDYN*	20.81	VFPS*	138.33	DEN*	.0021749

Non-Dimensional Derivative Matrix

Perturbation	CY	CL	CN
B (per deg)	-.01800000	-.00060000	.00290000
P (per rad)	0.00000000	-.45600000	-.02000000
R (per rad)	.75000000	.07500000	-.35000000
BD (per rad)	0.00000000	0.00000000	0.00000000
DA (per deg)	.00180000	.00420000	-.00021000
DR (per deg)	.00265000	.00015000	-.00135000

Dimensional Derivative Parameter Matrix*

Perturbation	Y	L	N
B (per rad)	-.92981E+02	-.15903E+02	.21644E+02
P (per rad)	0.	-.85145E+01	-.10516E+00
R (per rad)	.27293E+01	.14004E+01	-.18403E+01
BD (per rad)	0.	0.	0.
DA (per rad)	.92981E+01	.11132E+03	-.15673E+01
DR (per rad)	.13689E+02	.39757E+01	-.10076E+02

* Internally Calculated

Table A-7

Case III (Lateral Axis)

Flight Case Data

MACH	.12500	ALT	6000.	WGHT	135.
AOA	0.00	SPAN	11.17	AREA	18.18
IXX	9.	IZZ	32.	IXZ	-2.
QDYN*	18.60	VFPS*	136.86	DEN*	.0019863

Non-Dimensional Derivative Matrix

Perturbation	CY	CL	CN
B (per deg)	-.01800000	-.00060000	.00290000
P (per rad)	0.00000000	-.45600000	-.02000000
R (per rad)	.75000000	.07500000	-.35000000
BD (per rad)	0.00000000	0.00000000	0.00000000
DA (per deg)	.00180000	.00420000	-.00021000
DR (per deg)	.00265000	.00015000	-.00135000

Dimensional Derivative Parameter Matrix*

Perturbation	Y	L	N
B (per rad)	-.83130E+02	-.14218E+02	.19351E+02
P (per rad)	0.	-.76939E+01	-.95023E-01
R (per rad)	.24663E+01	.12654E+01	-.16629E+01
BD (per rad)	0.	0.	0.
DA (per rad)	.83130E+01	.99525E+02	-.14013E+01
DR (per rad)	.12239E+02	.35545E+01	-.90082E+01

* Internally Calculated

Table A-8

Dimensional Derivative Parameters

Longitudinal - Stability Axes

$X'_u = -\frac{\rho S V_T}{m}(C_D + C_{D_u})$	$Z'_u = -\frac{\rho S V_T}{m}(C_L + C_{L_u})$	$M'_u = \frac{\rho S V_T}{I_{YY}}(C_m + C_{m_u})$
$X'_w = \frac{\rho S V_T}{2m}(C_L - C_{D_\alpha})$	$Z'_w = -\frac{\rho S V_T}{2m}(C_D + C_{L_\alpha})$	$M'_w = \frac{\rho S V_T \bar{c}}{2 I_{YY}} C_{m_\alpha}$
$X'_q = -\frac{\rho S V_T \bar{c}}{4m} C_{D_q}$	$Z'_q = -\frac{\rho S V_T \bar{c}}{4m} C_{L_q}$	$M'_q = \frac{\rho S V_T \bar{c}^2}{4 I_{YY}} C_{m_q}$
$X'_{\dot{w}} = -\frac{\rho S \bar{c}}{4m} C_{D_{\dot{\alpha}}}$	$Z'_{\dot{w}} = -\frac{\rho S \bar{c}}{4m} C_{L_{\dot{\alpha}}}$	$M'_{\dot{w}} = \frac{\rho S \bar{c}^2}{4 I_{YY}} C_{m_{\dot{\alpha}}}$
$X'_{\delta e} = -\frac{\rho S V_T^2}{2m} C_{D_{\delta e}}$	$Z'_{\delta e} = -\frac{\rho S V_T^2}{2m} C_{L_{\delta e}}$	$M'_{\delta e} = \frac{\rho S V_T^2 \bar{c}}{2 I_{YY}} C_{m_{\delta e}}$

Non-Dimensional Derivatives Per Radian

Table A-9

Longitudinal Stability Axes to Body Axes Conversion

$$X_u = X'_u \cos^2\alpha - (X'_w + Z'_u)\sin\alpha\cos\alpha + Z'_w \sin^2\alpha$$
$$X_w = X'_w \cos^2\alpha + (X'_u - Z'_w)\sin\alpha\cos\alpha - Z'_u \sin^2\alpha$$
$$X_q = X'_q \cos\alpha - Z'_q \sin\alpha$$
$$X_{\dot{w}} = X'_{\dot{w}} \cos^2\alpha - Z'_{\dot{w}} \sin\alpha\cos\alpha$$
$$X_{\delta_e} = X'_{\delta_e} \cos\alpha - Z'_{\delta_e} \sin\alpha$$

$$Z_u = Z'_u \cos^2\alpha - (Z'_w - X'_u)\sin\alpha\cos\alpha - X'_w \sin^2\alpha$$
$$Z_w = Z'_w \cos^2\alpha + (Z'_u + X'_w)\sin\alpha\cos\alpha + X'_u \sin^2\alpha$$
$$Z_q = Z'_q \cos\alpha - X'_q \sin\alpha$$
$$Z_{\dot{w}} = Z_{\dot{w}} \cos^2\alpha + X'_{\dot{w}} \sin\alpha\cos\alpha$$
$$Z_{\delta_e} = Z'_{\delta_e} \cos\alpha + X'_{\delta_e} \sin\alpha$$

$$M_u = M'_u \cos\alpha - M'_w \sin\alpha$$
$$M_w = M'_w \cos\alpha + M'_u \sin\alpha$$
$$M_q = M'_q$$
$$M_{\dot{w}} = M'_{\dot{w}} \cos\alpha$$
$$M_{\delta_e} = M'_{\delta_e}$$

Table A-10

Dimensional Derivative Parameters

Lateral-Directional-Body Axes

$Y_\beta = \frac{\rho S V_T^2}{2m} C_{Y_\beta}$	$L_\beta = \frac{\rho S V_T^2 b}{2I_{xx}} C_{\ell_\beta}$	$N_\beta = \frac{\rho S V_T^2 b}{2I_{zz}} C_{n_\beta}$
$Y_p = \frac{\rho S V_T b}{4m} C_{Y_p}$	$L_p = \frac{\rho S V_T b^2}{4I_{xx}} C_{\ell_p}$	$N_p = \frac{\rho S V_T b^2}{4I_{zz}} C_{n_p}$
$Y_r = \frac{\rho S V_T b}{4m} C_{Y_r}$	$L_r = \frac{\rho S V_T b^2}{4I_{xx}} C_{\ell_r}$	$N_r = \frac{\rho S V_T b^2}{4I_{zz}} C_{n_r}$
$Y_{\dot\beta} = \frac{\rho S V_T b}{4m} C_{Y_{\dot\beta}}$	$L_{\dot\beta} = \frac{\rho S V_T b^2}{4I_{xx}} C_{\ell_{\dot\beta}}$	$N_{\dot\beta} = \frac{\rho S V_T b^2}{4I_{zz}} C_{n_{\dot\beta}}$
$Y_{\delta_a} = \frac{\rho S V_T^2}{2m} C_{Y_{\delta_a}}$	$L_{\delta_a} = \frac{\rho S V_T^2 b}{2I_{xx}} C_{\ell_{\delta_a}}$	$N_{\delta_a} = \frac{\rho S V_T^2 b}{2I_{zz}} C_{n_{\delta_a}}$
$Y_{\delta_r} = \frac{\rho S V_T^2}{2m} C_{Y_{\delta_r}}$	$L_{\delta_r} = \frac{\rho S V_T^2 b}{2I_{xx}} C_{\ell_{\delta_r}}$	$N_{\delta_r} = \frac{\rho S V_T^2 b}{2I_{zz}} C_{n_{\delta_r}}$

Table A-11

Lateral Stability Axes to Body Axes Conversion

$$C_{y_\beta} = C'_{y_\beta}$$

$$C_{y_p} = C'_{y_p} \cos\alpha - C'_{y_r} \sin\alpha$$

$$C_{y_r} = C'_{y_p} \sin\alpha + C'_{y_r} \cos\alpha$$

$$C_{y_{\dot\beta}} = C'_{y_{\dot\beta}}$$

$$C_{y_{\delta_a}} = C'_{y_{\delta_a}}$$

$$C_{y_{\delta_r}} = C'_{y_{\delta_r}}$$

$$C_{\ell_\beta} = C'_{\ell_\beta} \cos\alpha - C'_{n_\beta} \sin\alpha$$

$$C_{\ell_p} = C'_{\ell_p} \cos^2\alpha + C'_{n_r} \sin^2\alpha - (C'_{n_p} + C'_{\ell_r}) \sin\alpha \cos\alpha$$

$$C_{\ell_r} = C'_{\ell_r} \cos^2\alpha - C'_{n_p} \sin^2\alpha + (C'_{\ell_p} - C'_{n_r}) \sin\alpha \cos\alpha$$

$$C_{\ell_{\dot\beta}} = C'_{\ell_{\dot\beta}} \cos\alpha - C'_{n_{\dot\beta}} \sin\alpha$$

$$C_{\ell_{\delta_a}} = C'_{\ell_{\delta_a}} \cos\alpha - C'_{n_{\delta_a}} \sin\alpha$$

$$C_{\ell_{\delta_r}} = C'_{\ell_{\delta_r}} \cos\alpha - C'_{n_{\delta_r}} \sin\alpha$$

$$C_{n_\beta} = C'_{\ell_\beta} \sin\alpha + C'_{n_\beta} \cos\alpha$$

$$C_{n_p} = C'_{n_p} \cos^2\alpha - C'_{\ell_r} \sin^2\alpha + (C'_{\ell_p} - C'_{n_r}) \sin\alpha \cos\alpha$$

$$C_{n_r} = C'_{n_r} \cos^2\alpha + C'_{\ell_p} \sin^2\alpha + (C'_{\ell_r} + C'_{n_p}) \sin\alpha \cos\alpha$$

$$C_{n_{\dot\beta}} = C'_{\ell_{\dot\beta}} \sin\alpha + C'_{n_{\dot\beta}} \cos\alpha$$

$$C_{n_{\delta_a}} = C'_{\ell_{\delta_a}} \sin\alpha + C'_{n_{\delta_a}} \cos\alpha$$

$$C_{n_{\delta_r}} = C'_{\ell_{\delta_r}} \sin\alpha + C'_{n_{\delta_r}} \cos\alpha$$

Note: All primed coefficients appear as input data in the program TTYLAT.

Table A-12

Longitudinal RPV Transfer Functions

Transfer Function	Standard Form	Bode Form	Case
$-\frac{\theta(S)}{\delta_e(S)}$, TH/DE	$\frac{(64.634)(S+.09293)(S+3.4269)}{(S^2+.0611S+.1074)(S^2+7.0324S+53.8064)}$	$\frac{(3.562)\left(\frac{S}{.09293}+1\right)\left(\frac{S}{3.4269}+1\right)}{\left[\left(\frac{S}{.3277}\right)^2+\frac{2(.09)}{.3277}S+1\right]\left[\left(\frac{S}{7.3353}\right)^2+\frac{2(.48)}{7.3353}S+1\right]}$	I
	$\frac{(60.079)(S+.09038)(S+3.2043)}{(S^2+.0570S+.1087)(S^2+6.5838S+49.5449)}$	$\frac{(3.231)\left(\frac{S}{.09038}+1\right)\left(\frac{S}{3.2043}+1\right)}{\left[\left(\frac{S}{.3297}\right)^2+\frac{2(.086)}{.3297}S+1\right]\left[\left(\frac{S}{7.0388}\right)^2+\frac{2(.468)}{7.0388}S+1\right]}$	II
	$\frac{(53.734)(S+.08735)(S+2.8915)}{(S^2+.0505S+.1182)(S^2+5.9790S+42.1424)}$	$\frac{(2.7246)\left(\frac{S}{.08735}+1\right)\left(\frac{S}{2.8925}+1\right)}{\left[\left(\frac{S}{.3438}\right)^2+\frac{2(.073)}{.3438}S+1\right]\left[\left(\frac{S}{6.4917}\right)^2+\frac{2(.461)}{6.4917}S+1\right]}$	III

Transfer Function	Standard Form	Bode Form	Case
$-\frac{\alpha(S)}{\delta_e(S)}$, AOA/DE	$\frac{(.4766)(S+134.32)(S^2+.0677S+.1073)}{(S^2+.0611S+.1074)(S^2+7.0324S+53.8064)}$	$\frac{(1.1887)\left(\frac{S}{134.32}+1\right)\left[\left(\frac{S}{.3276}\right)^2+\frac{2(.103)}{.3276}S+1\right]}{\left[\left(\frac{S}{.3277}\right)^2+\frac{2(.09)}{.3277}S+1\right]\left[\left(\frac{S}{7.3353}\right)^2+\frac{2(.48)}{7.3353}S+1\right]}$	I
	$\frac{(.4462)(S+133.44)(S^2+.0635S+.1048)}{(S^2+.0570S+.1087)(S^2+6.5838S+49.5449)}$	$\frac{(1.1586)\left(\frac{S}{133.44}+1\right)\left[\left(\frac{S}{.3237}\right)^2+\frac{2(.0981)}{.3237}S+1\right]}{\left[\left(\frac{S}{.3297}\right)^2+\frac{2(.086)}{.3297}S+1\right]\left[\left(\frac{S}{7.0388}\right)^2+\frac{2(.468)}{7.0388}S+1\right]}$	II
	$\frac{(.4034)(S+132.10)(S^2+.0577S+.1014)}{(S^2+.0505S+.1182)(S^2+5.9790S+42.1424)}$	$\frac{(1.0848)\left(\frac{S}{132.10}+1\right)\left[\left(\frac{S}{.3184}\right)^2+\frac{2(.0906)}{.3184}S+1\right]}{\left[\left(\frac{S}{.3438}\right)^2+\frac{2(.073)}{.3438}S+1\right]\left[\left(\frac{S}{6.4917}\right)^2+\frac{2(.461)}{6.4917}S+1\right]}$	III

Transfer Function	Standard Form	Bode Form	Case
$-\frac{U(S)}{\delta_e(S)}$, U/DE	$\frac{(.9506)(S-31.936)(S^2+24.978S+236.4683)}{(S^2+.0611S+.1074)(S^2+7.0324S+53.8064)}$	$\frac{(1242.26)\left(\frac{S}{31.936}-1\right)\left[\left(\frac{S}{15.38}\right)^2+\frac{2(.812)}{15.38}S+1\right]}{\left[\left(\frac{S}{.3277}\right)^2+\frac{2(.09)}{.3277}S+1\right]\left[\left(\frac{S}{7.3353}\right)^2+\frac{2(.48)}{7.3353}S+1\right]}$	I
	$\frac{(.7433)(S-75.223)(S^2+11.1238S+111.7728)}{(S^2+.0570S+.1087)(S^2+6.5838S+49.5449)}$	$\frac{(1160.44)\left(\frac{S}{75.223}-1\right)\left[\left(\frac{S}{10.57}\right)^2+\frac{2(.526)}{10.57}S+1\right]}{\left[\left(\frac{S}{.3297}\right)^2+\frac{2(.086)}{.3297}S+1\right]\left[\left(\frac{S}{7.0388}\right)^2+\frac{2(.468)}{7.0388}S+1\right]}$	II
	$\frac{(.4543)(S-233.97)(S^2+5.2444S+47.5969)}{(S^2+.0505S+.1182)(S^2+5.9790S+42.1424)}$	$\frac{(1015.65)\left(\frac{S}{233.97}-1\right)\left[\left(\frac{S}{6.90}\right)^2+\frac{2(.38)}{6.90}S+1\right]}{\left[\left(\frac{S}{.3438}\right)^2+\frac{2(.073)}{.3438}S+1\right]\left[\left(\frac{S}{6.4917}\right)^2+\frac{2(.461)}{6.4917}S+1\right]}$	III

NOTE:

$\dot{\theta}(S)/\delta_e(S) = \theta(S)/\delta_e(S) \cdot (S)$ (THDOT/DE)

Table A-13

Lateral RPV Transfer Functions

Transfer Function	Standard Form	Bode Form	Case
$\frac{\phi(S)}{\delta_a(S)}$, PHI/DA	$\frac{(121.09)(S^2+2.6464S+23.9076)}{(S-.0012215)(S+9.1941)(S^2+2.7372S+24.9275)}$	$\frac{(10341.01)\left[\left(\frac{S}{4.8895}\right)^2+\frac{2(.271)}{4.8896}S+1\right]}{\left(\frac{S}{.0012215}-1\right)\left(\frac{S}{9.1941}+1\right)\left[\left(\frac{S}{4.9927}\right)^2+\frac{2(.274)}{4.9927}S+1\right]}$	I
	$\frac{(112.52)(S^2+2.4766S+22.1686)}{(S-.0012306)(S+8.6059)(S^2+2.5592S+23.1466)}$	$\frac{(10173.96)\left[\left(\frac{S}{4.7084}\right)^2+\frac{2(.263)}{4.7084}S+1\right]}{\left(\frac{S}{.0012306}-1\right)\left(\frac{S}{8.6059}+1\right)\left[\left(\frac{S}{4.8111}\right)^2+\frac{2(.266)}{4.8111}S+1\right]}$	II
	$\frac{(100.6)(S^2+2.2378S+19.7574)}{(S-.001244)(S+7.7809)(S^2+2.3082S+20.6761)}$	$\frac{(9931.36)\left[\left(\frac{S}{4.445}\right)^2+\frac{2(.252)}{4.445}S+1\right]}{\left(\frac{S}{.001244}-1\right)\left(\frac{S}{7.7809}+1\right)\left[\left(\frac{S}{4.5471}\right)^2+\frac{2(.254)}{4.5471}S+1\right]}$	III

NOTE:

$\dot{\phi}(S)/\delta_a(S) = \phi(S)/\delta_a(S) \cdot (S)$ (PHIDOT/DA)

Transfer Function	Standard Form	Bode Form	Case
$-\frac{\psi(S)}{\delta_a(S)}$, PSI/DA	$\frac{(7.545)(S-3.296)(S^2+7.6324S+25.865)}{(S-.0012215)(S+9.1941)(S^2+2.7372S+24.9275)(S)}$	$\frac{(2297.6)\left(\frac{S}{3.296}-1\right)\left[\left(\frac{S}{5.086}\right)^2+\frac{2(.75)}{5.086}S+1\right]}{S\left(\frac{S}{.0012215}-1\right)\left(\frac{S}{9.1941}+1\right)\left[\left(\frac{S}{4.9927}\right)^2+\frac{2(.274)}{4.9927}S+1\right]}$	I
	$\frac{(7.0113)(S-3.259)(S^2+7.3176S+24.474)}{(S-.0012306)(S+8.6059)(S^2+2.5592S+23.1466)(S)}$	$\frac{(2281.3)\left(\frac{S}{3.259}-1\right)\left[\left(\frac{S}{4.95}\right)^2+\frac{2(.74)}{4.95}S+1\right]}{S\left(\frac{S}{.0012306}-1\right)\left(\frac{S}{8.6059}+1\right)\left[\left(\frac{S}{4.8111}\right)^2+\frac{2(.266)}{4.8111}S+1\right]}$	II
	$\frac{(6.2685)(S-3.202)(S^2+6.8696S+22.508)}{(S-.001244)(S+7.7809)(S^2+2.3082S+20.6761)(S)}$	$\frac{(2257.4)\left(\frac{S}{3.202}-1\right)\left[\left(\frac{S}{4.74}\right)^2+\frac{2(.72)}{4.74}S+1\right]}{S\left(\frac{S}{.001244}-1\right)\left(\frac{S}{7.7809}+1\right)\left[\left(\frac{S}{4.5471}\right)^2+\frac{2(.254)}{4.5471}S+1\right]}$	III

NOTE:

$\dot{\psi}(S)/\delta_a(S) = \psi(S)/\delta_a(S) \cdot (S)$ (PSIDOT/DA)

Transfer Function	Standard Form	Bode Form	Case
$\frac{\beta(S)}{\delta_a(S)}$, BETA/DA	$\frac{(.07183)(S+1.1115)(S+6.3575)(S+106.58)}{(S-.0012215)(S+9.1941)(S^2+2.7372S+24.9275)}$	$\frac{(193.2)\left(\frac{S}{1.1115}+1\right)\left(\frac{S}{6.3575}+1\right)\left(\frac{S}{106.58}+1\right)}{\left(\frac{S}{.0012215}-1\right)\left(\frac{S}{9.1941}+1\right)\left[\left(\frac{S}{4.9927}\right)^2+\frac{2(.274)}{4.9927}S+1\right]}$	I
	$\frac{(.06722)(S+1.0833)(S+6.1889)(S+105.47)}{(S-.0012306)(S+8.6059)(S^2+2.5592S+23.1466)}$	$\frac{(193.9)\left(\frac{S}{1.0833}+1\right)\left(\frac{S}{6.1889}+1\right)\left(\frac{S}{105.47}+1\right)}{\left(\frac{S}{.0012306}-1\right)\left(\frac{S}{8.6059}+1\right)\left[\left(\frac{S}{4.8111}\right)^2+\frac{2(.266)}{4.8111}S+1\right]}$	II
	$\frac{(.06074)(S+1.0323)(S+5.9615)(S+103.83)}{(S-.001244)(S+7.7809)(S^2+2.3082S+20.6761)}$	$\frac{(193.9)\left(\frac{S}{1.0323}+1\right)\left(\frac{S}{5.9615}+1\right)\left(\frac{S}{103.83}+1\right)}{\left(\frac{S}{.001244}-1\right)\left(\frac{S}{7.7809}+1\right)\left[\left(\frac{S}{4.5471}\right)^2+\frac{2(.254)}{4.5471}S+1\right]}$	III

Table A-13 (Continued)

Lateral RPV Transfer Functions (Continued)

Transfer Function	Standard Form	Bode Form	Case
$-\frac{\psi(S)}{\delta_r(S)}$, PSI/DR	$\frac{(11.142)(S+8.9926)(S^2+.46686S+.1997)}{(S-.0012215)(S+9.1941)(S^2+2.7372S+24.9275)(S)}$	$\frac{(71.47)\left(\frac{S}{8.9926}+1\right)\left[\left(\frac{S}{.447}\right)^2+\frac{2(.52)}{.447}S+1\right]}{\left(\frac{S}{.0012215}-1\right)\left(\frac{S}{9.1941}+1\right)\left[\left(\frac{S}{4.9927}\right)^2+\frac{2(.274)}{4.9927}S+1\right]}$	I
	$\frac{(10.354)(S+8.4181)(S^2+.43378S+.1996)}{(S-.0012306)(S+8.6059)(S^2+2.5592S+23.1466)(S)}$	$\frac{(70.97)\left(\frac{S}{8.4181}+1\right)\left[\left(\frac{S}{.447}\right)^2+\frac{2(.49)}{.447}S+1\right]}{\left(\frac{S}{.0012306}-1\right)\left(\frac{S}{8.6059}+1\right)\left[\left(\frac{S}{4.8111}\right)^2+\frac{2(.266)}{4.8111}S+1\right]}$	II
	$\frac{(9.2571)(S+7.6118)(S^2+.3869S+.1995)}{(S-.001244)(S+7.7809)(S^2+2.3082S+20.6761)(S)}$	$\frac{(70.24)\left(\frac{S}{7.6118}+1\right)\left[\left(\frac{S}{.447}\right)^2+\frac{2(.43)}{.447}S+1\right]}{\left(\frac{S}{.001244}-1\right)\left(\frac{S}{7.7809}+1\right)\left[\left(\frac{S}{4.5471}\right)^2+\frac{2(.254)}{4.5471}S+1\right]}$	III

NOTE:

$\dot{\phi}(S)/\delta_r(S) = \psi(S)/\delta_r(S) \cdot (S)$ (PSIDOT/DR)

Transfer Function	Standard Form	Bode Form	Case
$\frac{\phi(S)}{\delta_r(S)}$, PHI/DR	$\frac{(6.1927)(S-4.3056)(S+3.3882)}{(S-.0012215)(S+9.1941)(S^2+2.7372S+24.9275)}$	$\frac{(322.7)\left(\frac{S}{4.3056}-1\right)\left(\frac{S}{3.3882}+1\right)}{\left(\frac{S}{.0012215}-1\right)\left(\frac{S}{9.1941}+1\right)\left[\left(\frac{S}{4.9927}\right)^2+\frac{2(.274)}{4.9927}S+1\right]}$	I
	$\frac{(5.7546)(S-4.1318)(S+3.2732)}{(S-.0012306)(S+8.6059)(S^2+2.5592S+23.1466)}$	$\frac{(317.5)\left(\frac{S}{4.1318}-1\right)\left(\frac{S}{3.2732}+1\right)}{\left(\frac{S}{.0012306}-1\right)\left(\frac{S}{8.6059}+1\right)\left[\left(\frac{S}{4.8111}\right)^2+\frac{2(.266)}{4.8111}S+1\right]}$	II
	$\frac{(5.145)(S-3.8809)(S+3.1051)}{(S-.001244)(S+7.7809)(S^2+2.3082S+20.6761)}$	$\frac{(309.8)\left(\frac{S}{3.8809}-1\right)\left(\frac{S}{3.1051}+1\right)}{\left(\frac{S}{.001244}-1\right)\left(\frac{S}{7.7809}+1\right)\left[\left(\frac{S}{4.5471}\right)^2+\frac{2(.254)}{4.5471}S+1\right]}$	III

NOTE:

$\dot{\phi}(S)/\delta_r(S) = \phi(S)/\delta_r(S) \cdot (S)$ (PHIDOT/DR)

Transfer Function	Standard Form	Bode Form	Case
$\frac{\beta(S)}{\delta_r(S)}$, BETA/DR	$\frac{(.10575)(S-.017937)(S+9.1125)(S+105.27)}{(S-.0012215)(S+9.1941)(S^2+2.7372S+24.9275)}$	$\frac{(6.5)\left(\frac{S}{.017937}-1\right)\left(\frac{S}{9.1125}+1\right)\left(\frac{S}{105.27}+1\right)}{\left(\frac{S}{.0012215}-1\right)\left(\frac{S}{9.1941}+1\right)\left[\left(\frac{S}{4.9927}\right)^2+\frac{2(.274)}{4.9927}S+1\right]}$	I
	$\frac{(.09896)(S-.018039)(S+8.5389)(S+104.54)}{(S-.0012306)(S+8.6059)(S^2+2.5592S+23.1466)}$	$\frac{(6.5)\left(\frac{S}{.018039}-1\right)\left(\frac{S}{8.5389}+1\right)\left(\frac{S}{104.54}+1\right)}{\left(\frac{S}{.0012306}-1\right)\left(\frac{S}{8.6059}+1\right)\left[\left(\frac{S}{4.8111}\right)^2+\frac{2(.266)}{4.8111}S+1\right]}$	II
	$\frac{(.08942)(S-.018192)(S+7.7333)(S+103.42)}{(S-.001244)(S+7.7809)(S^2+2.3082S+20.6761)}$	$\frac{(6.5)\left(\frac{S}{.018192}-1\right)\left(\frac{S}{7.7333}+1\right)\left(\frac{S}{103.42}+1\right)}{\left(\frac{S}{.001244}-1\right)\left(\frac{S}{7.7809}+1\right)\left[\left(\frac{S}{4.5471}\right)^2+\frac{2(.254)}{4.5471}S+1\right]}$	III

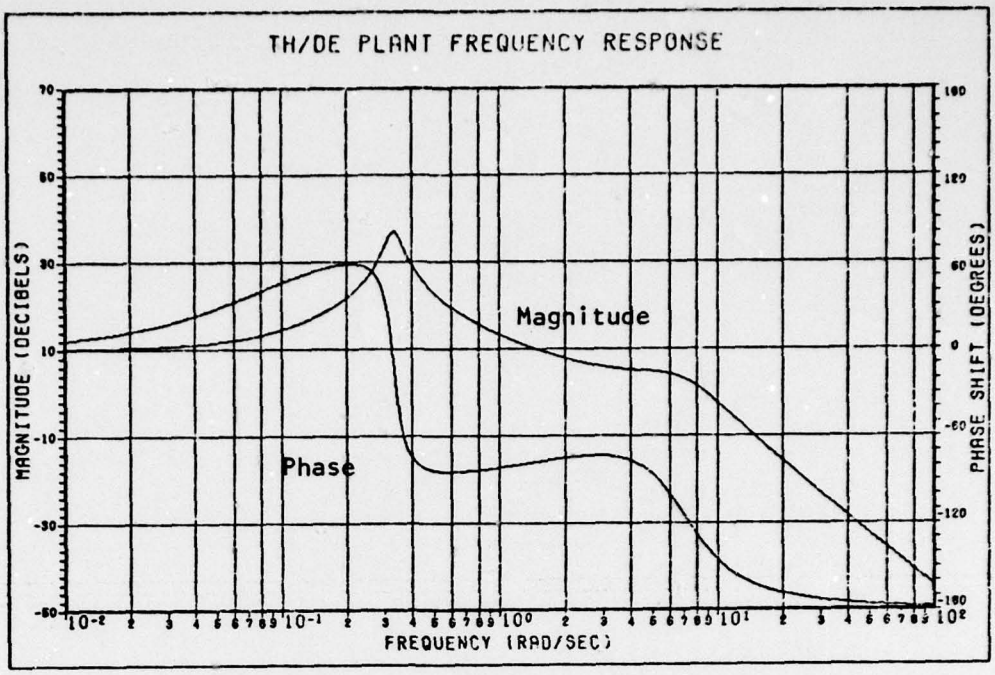


Figure A-1a: $-\theta/\delta_e$ Frequency Response

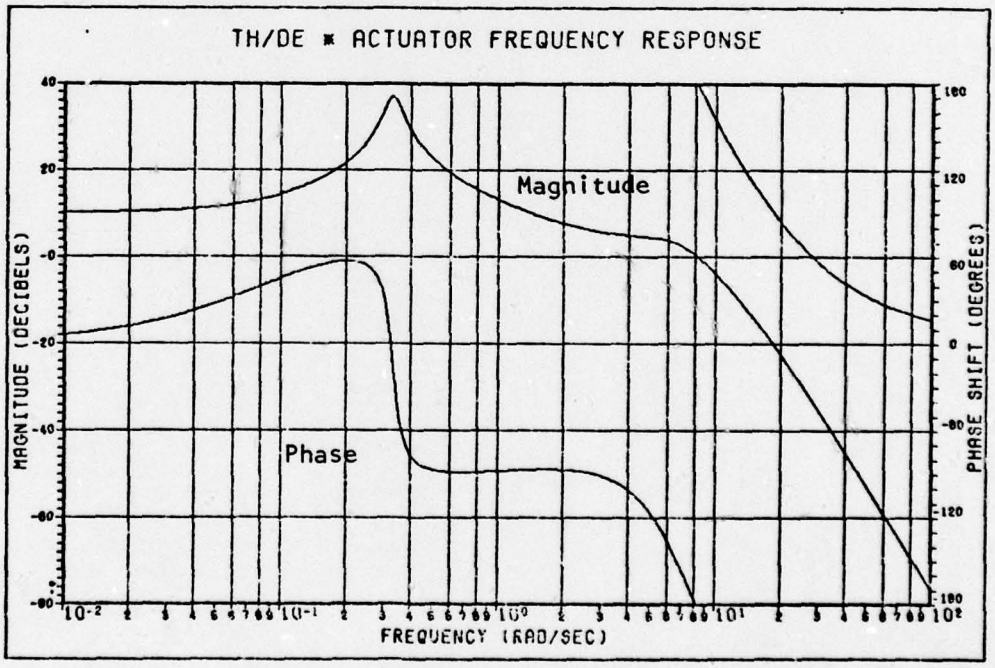


Figure A-1b: θ/δ_e * Actuator Frequency Response

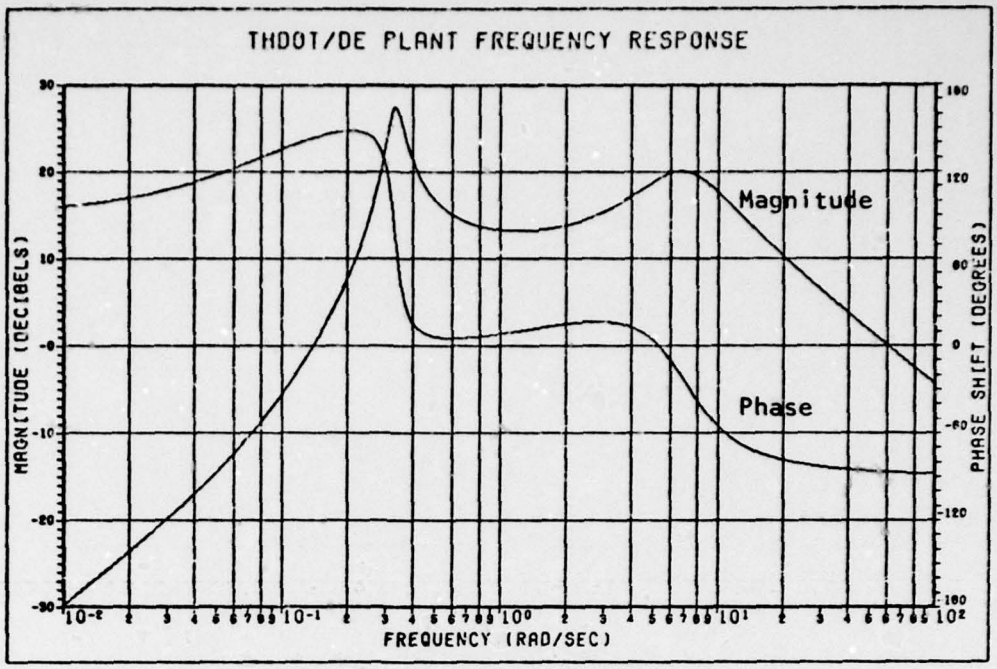


Figure A-2a: $-\dot{\theta}/\delta_e$ Frequency Response

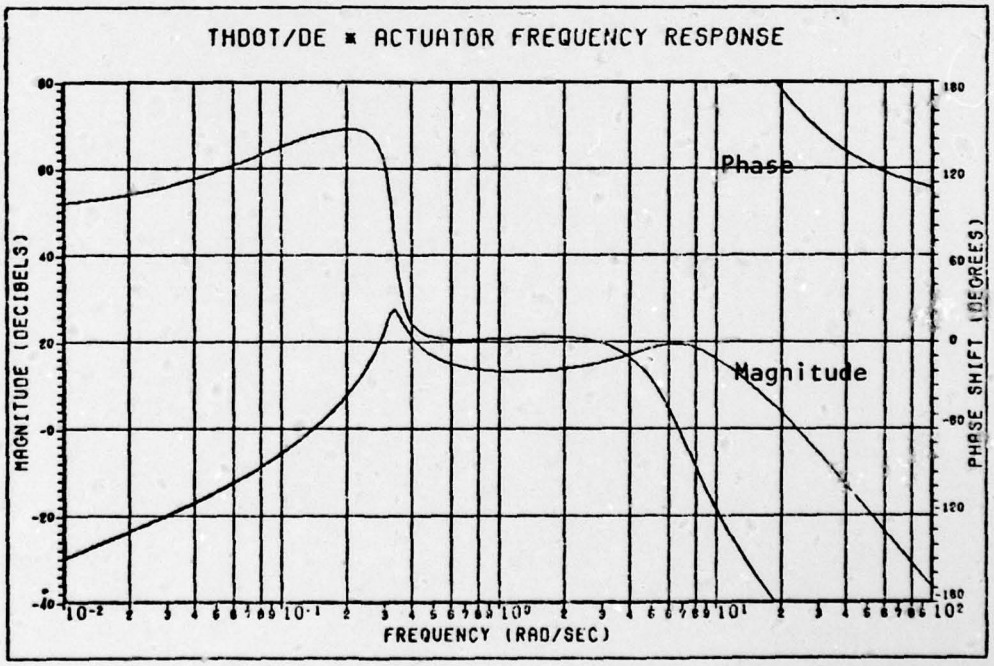


Figure A-2b: $\dot{\theta}/\delta_e$ * Actuator Frequency Response

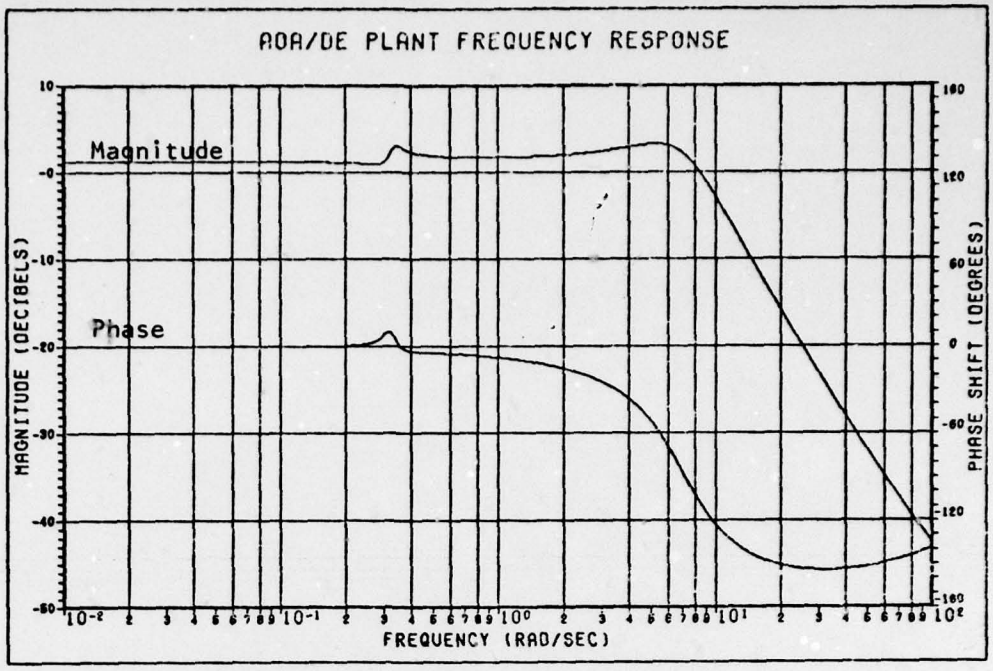


Figure A-3a: $-\alpha/\delta_e$ Frequency Response

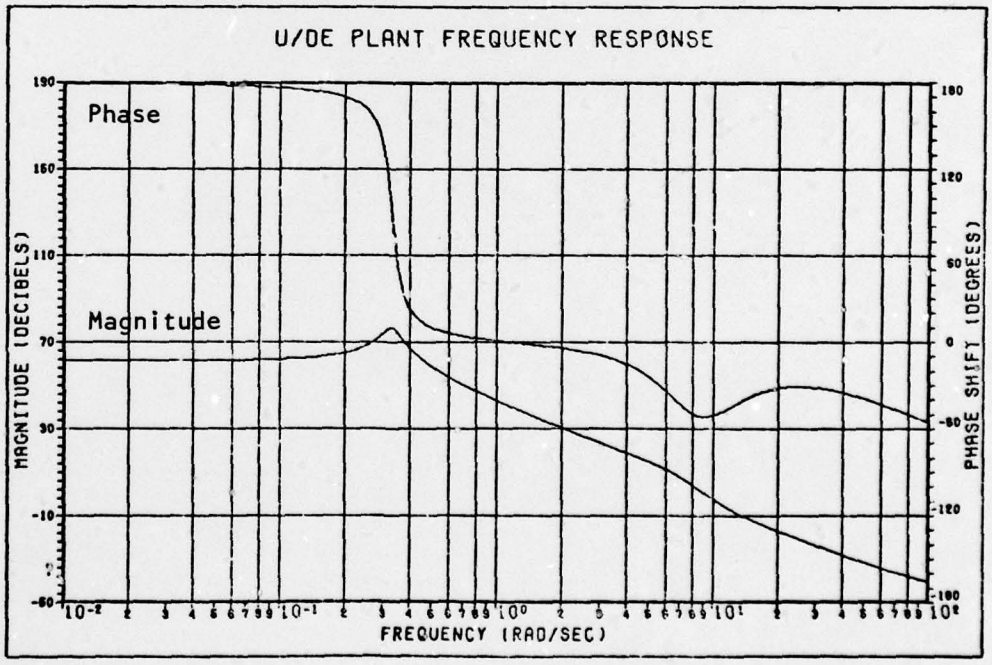


Figure A-3b: $-U/\delta_e$ Frequency Response

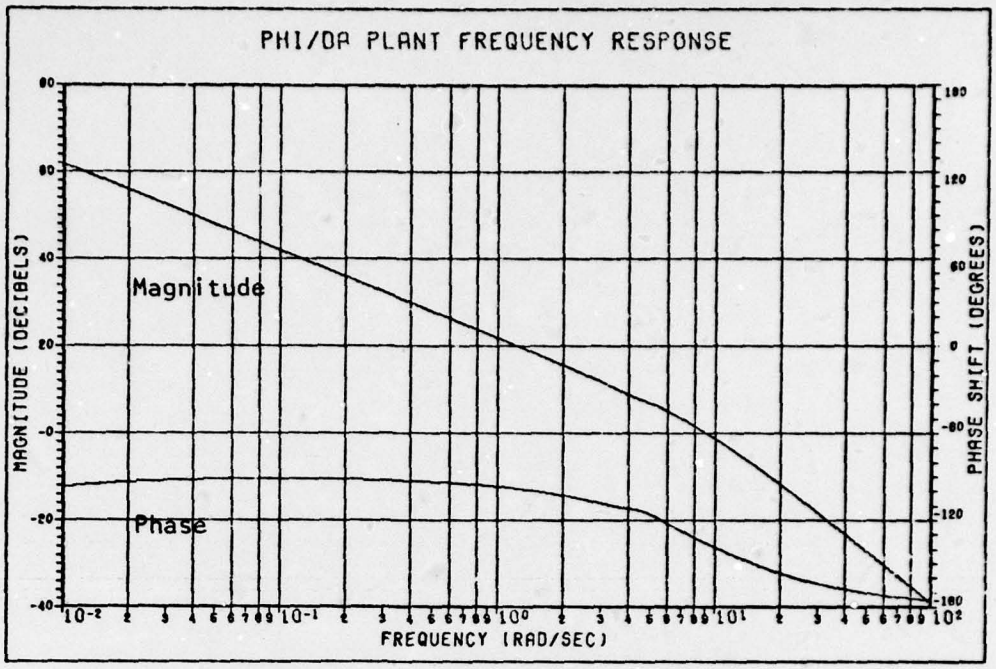


Figure A-4a: ϕ/δ_a Frequency Response

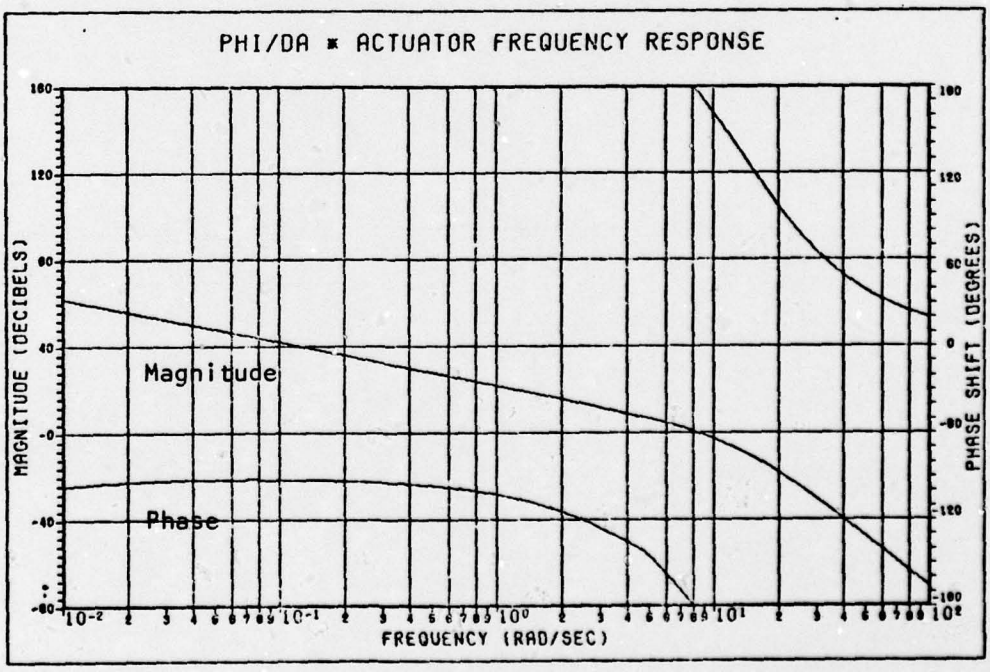


Figure A-4b: ϕ/δ_a * Actuator Frequency Response

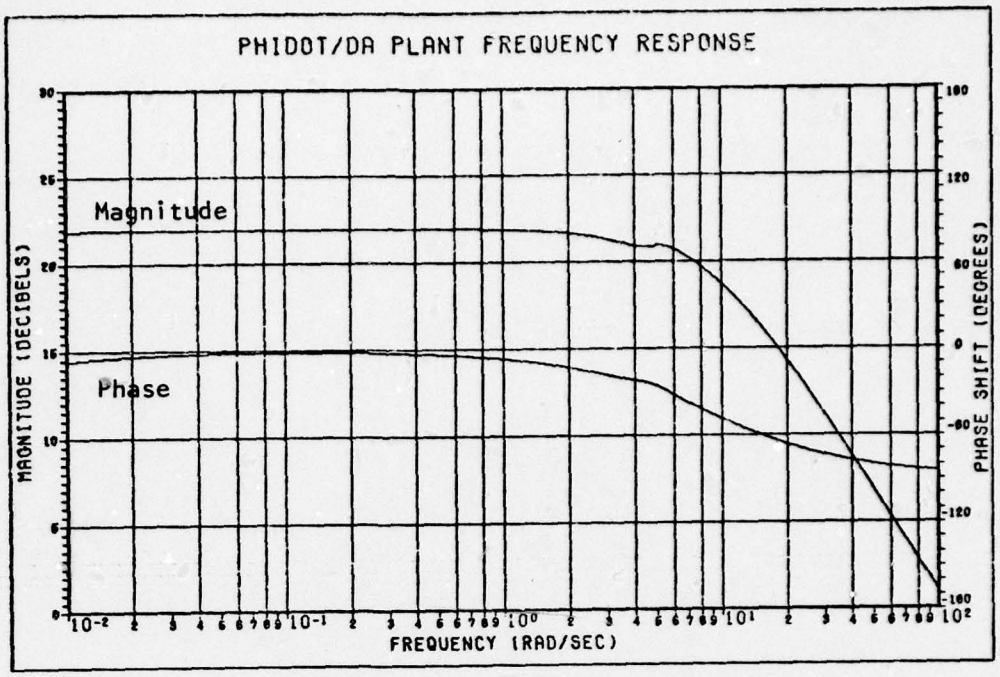


Figure A-5a: $\dot{\phi}/\delta_a$ Frequency Response

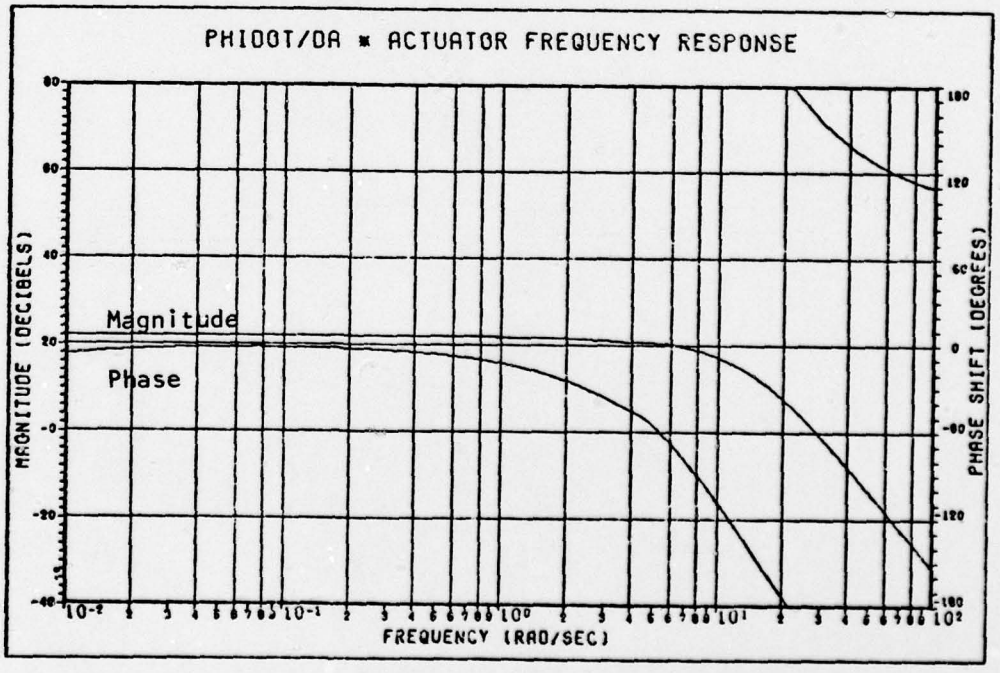


Figure A-5b: $\dot{\phi}/\delta_a$ * Actuator Frequency Response

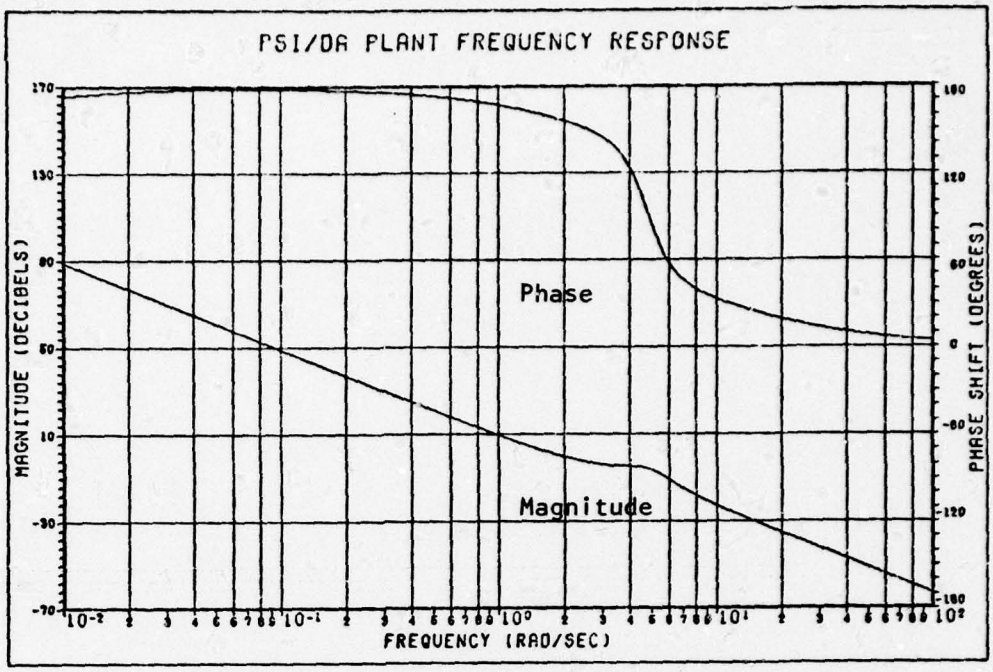


Figure A-6a: ψ/δ_a Frequency Response

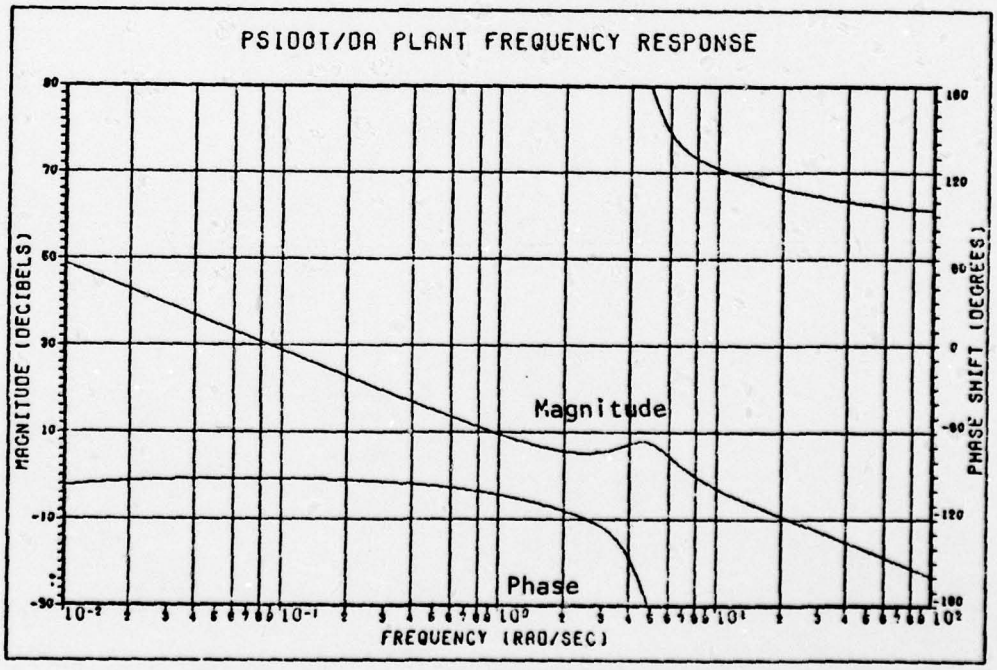


Figure A-6b: $\dot{\psi}/\delta_a$ Frequency Response

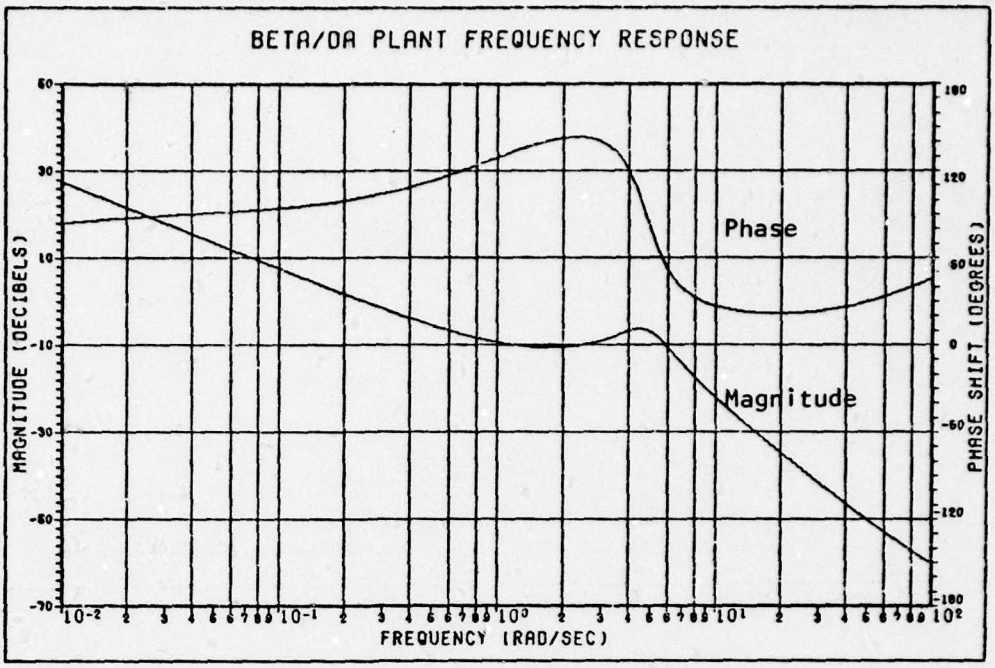


Figure A-7: β/δ_a Frequency Response

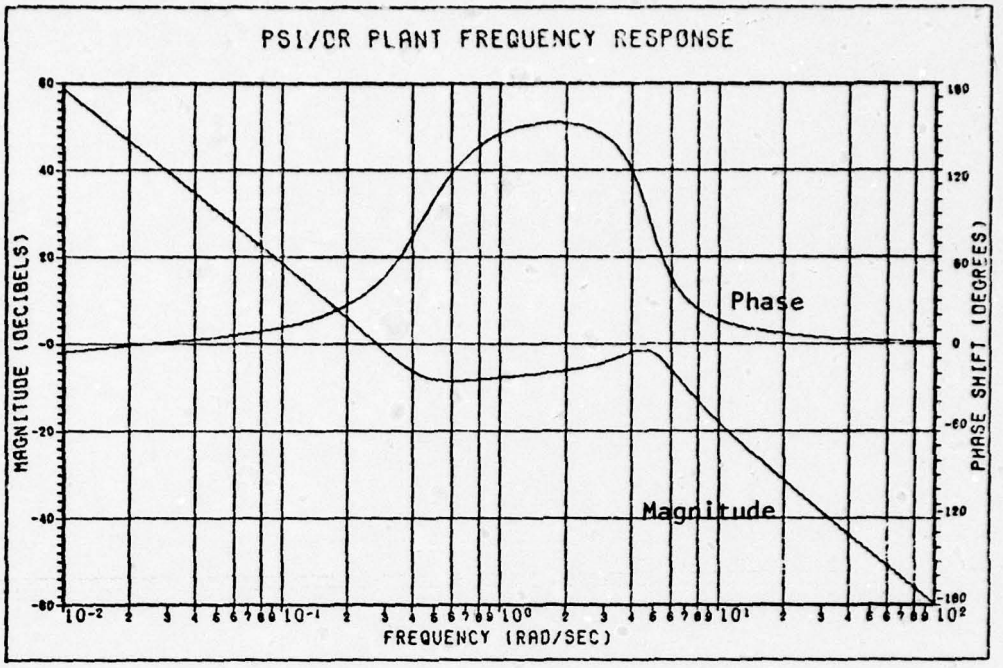


Figure A-8a: ψ/δ_r Frequency Response

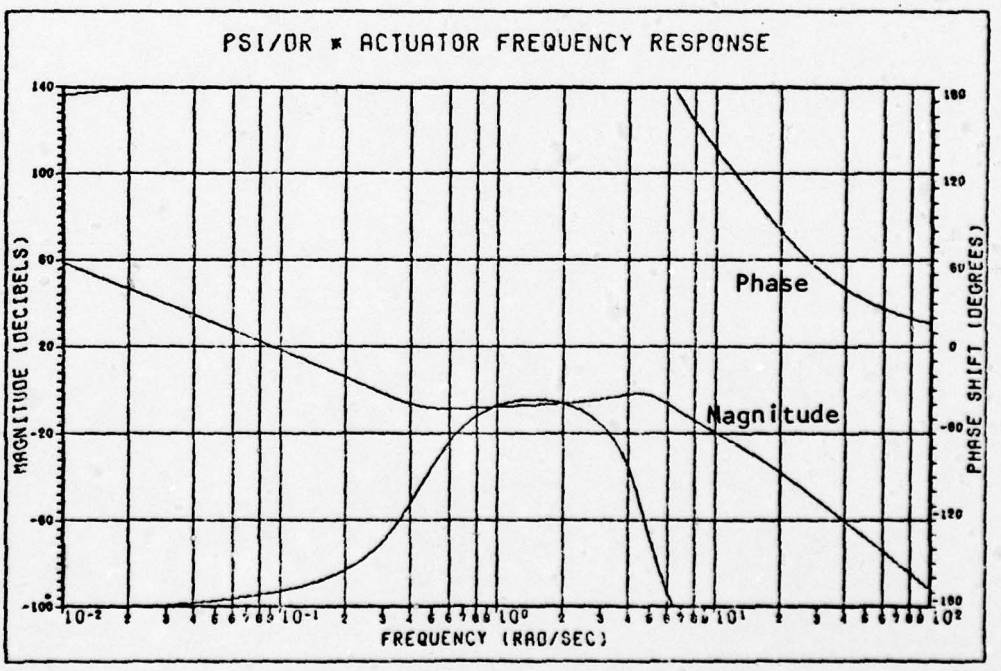


Figure A-8b: $-\psi/\delta_r$ * Actuator Frequency Response

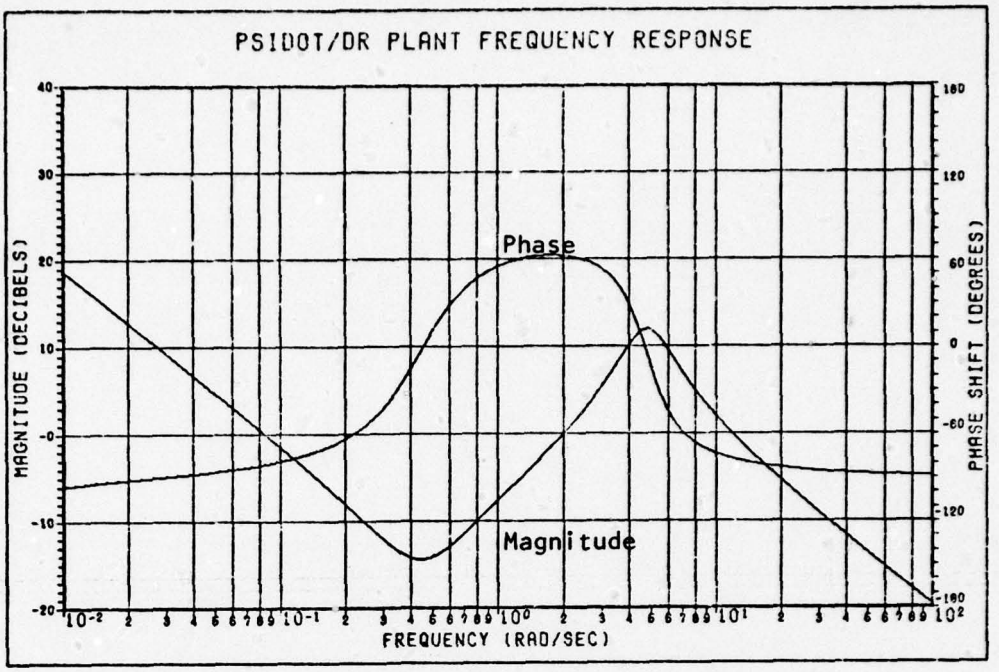


Figure A-9a: $\dot{\psi}/\delta_r$ Frequency Response

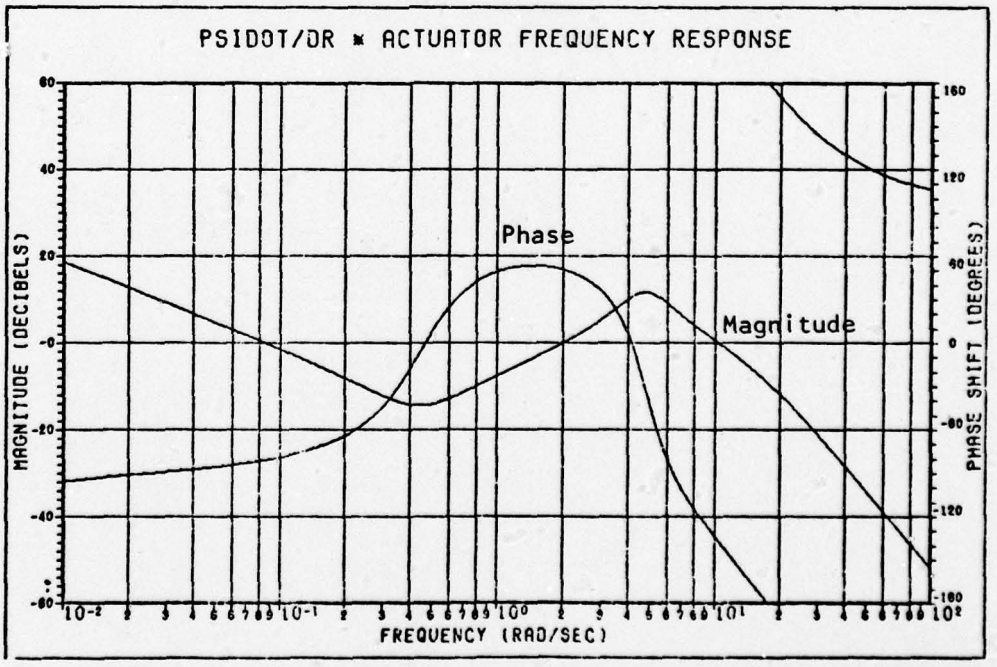


Figure A-9b: $-\dot{\psi}/\delta_r$ * Actuator Frequency Response

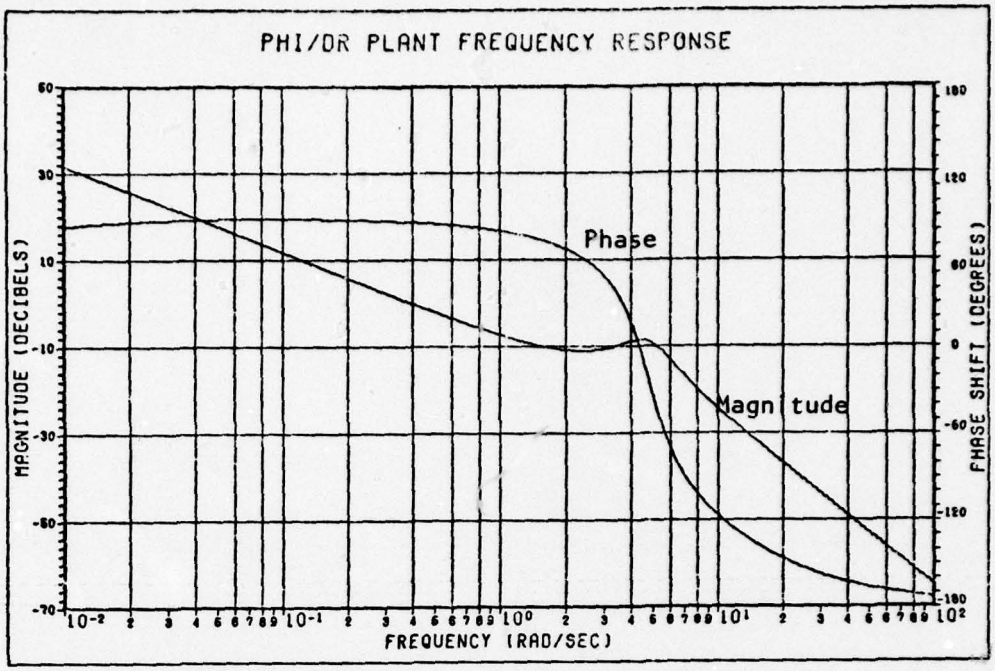


Figure A-10a: ϕ/δ_r Frequency Response

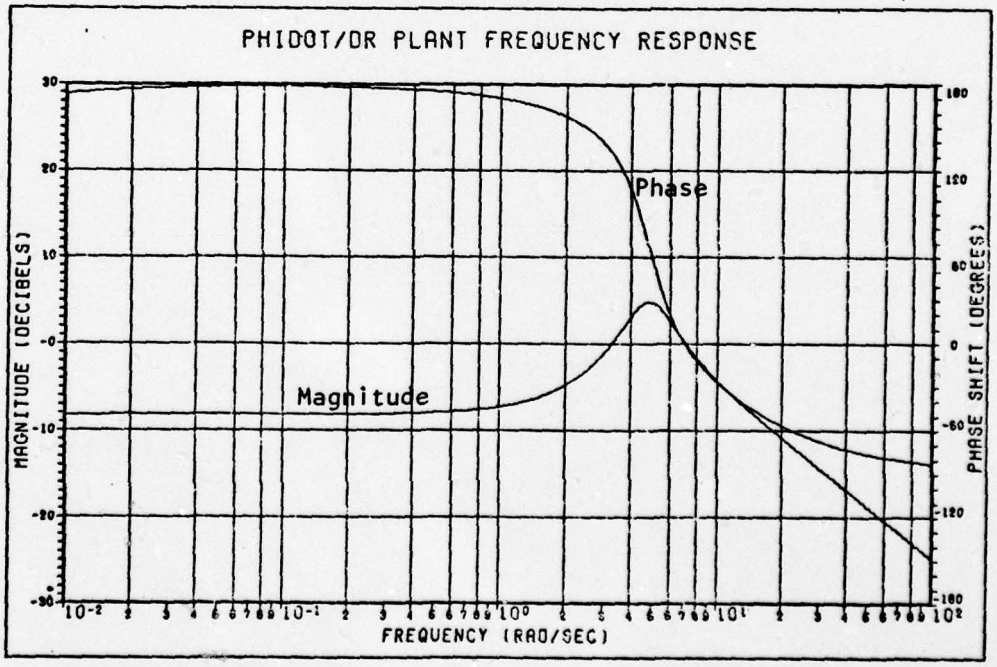


Figure A-10b: $\dot{\phi}/\delta_r$ Frequency Response

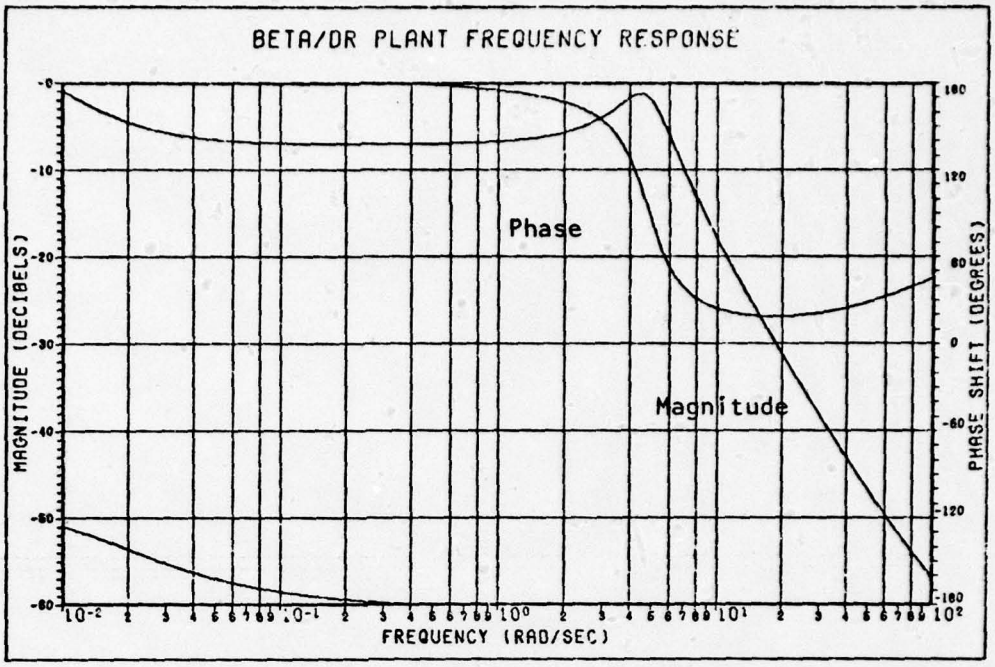


Figure A-11: β/δ_r Frequency Response

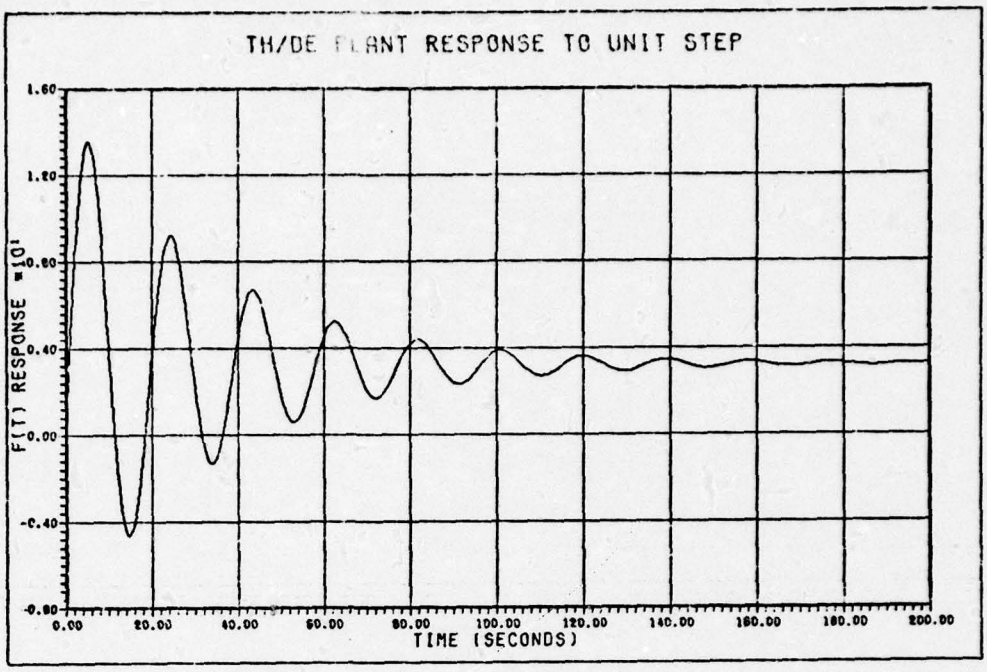


Figure A-12a: $-\theta/\delta_e$ Time Response

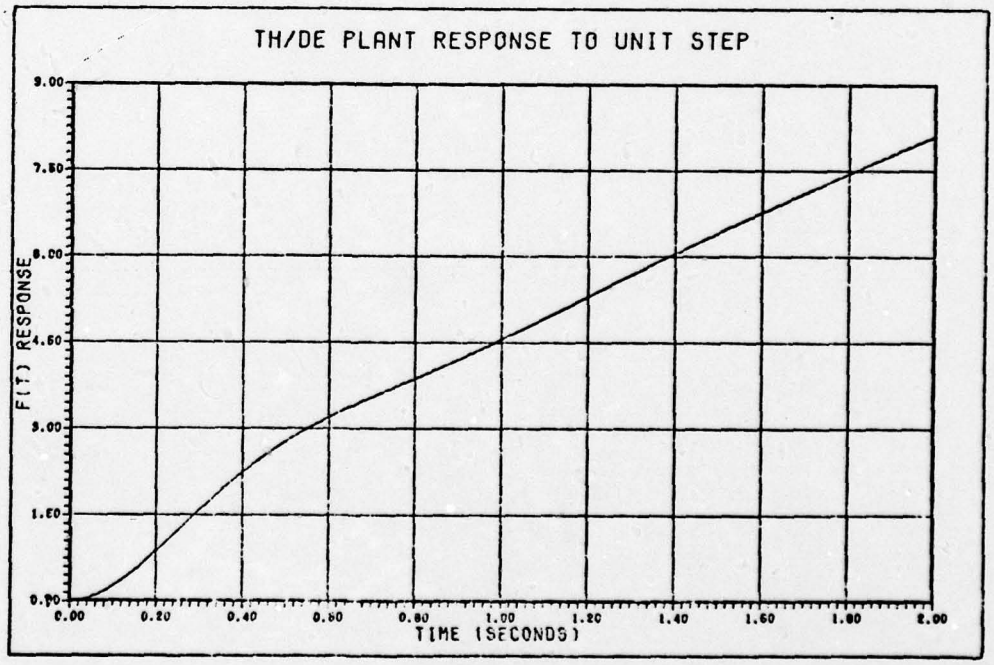


Figure A-12b: $-\theta/\delta_e$ Time Response (Phugoid)

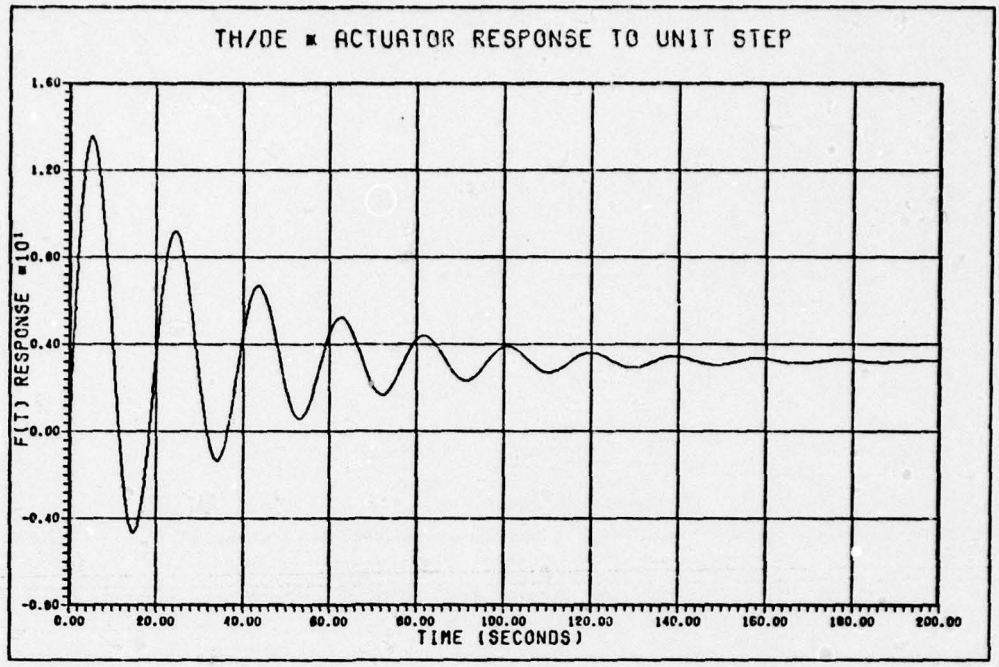


Figure A-13a: θ/δ_e * Actuator Time Response

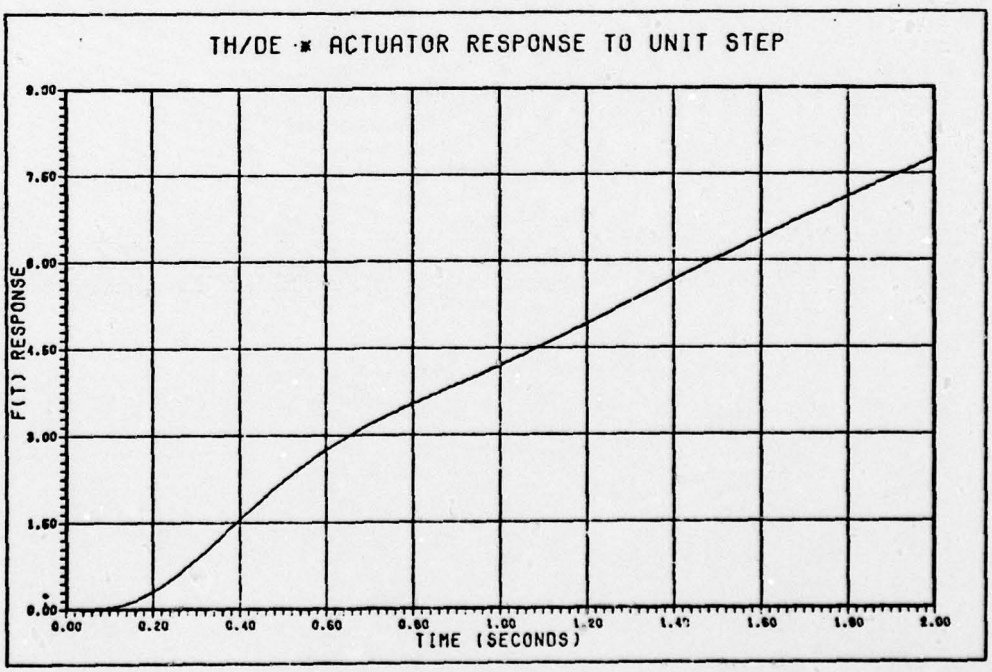


Figure A-13b: θ/δ_e * Actuator Time Response

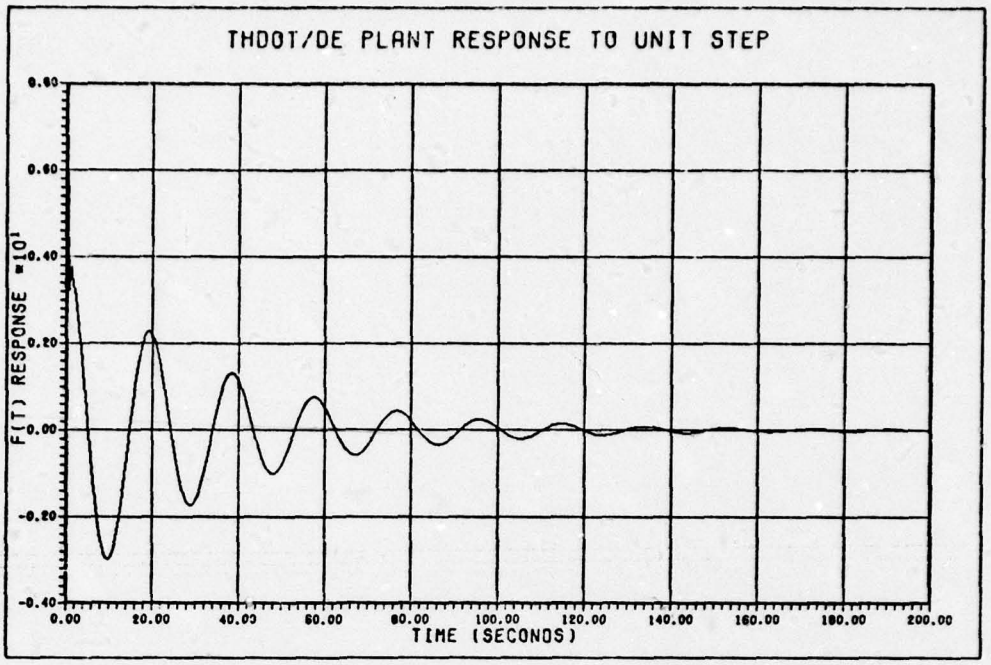


Figure A-14a: $-\dot{\theta}/\delta_e$ Time Response

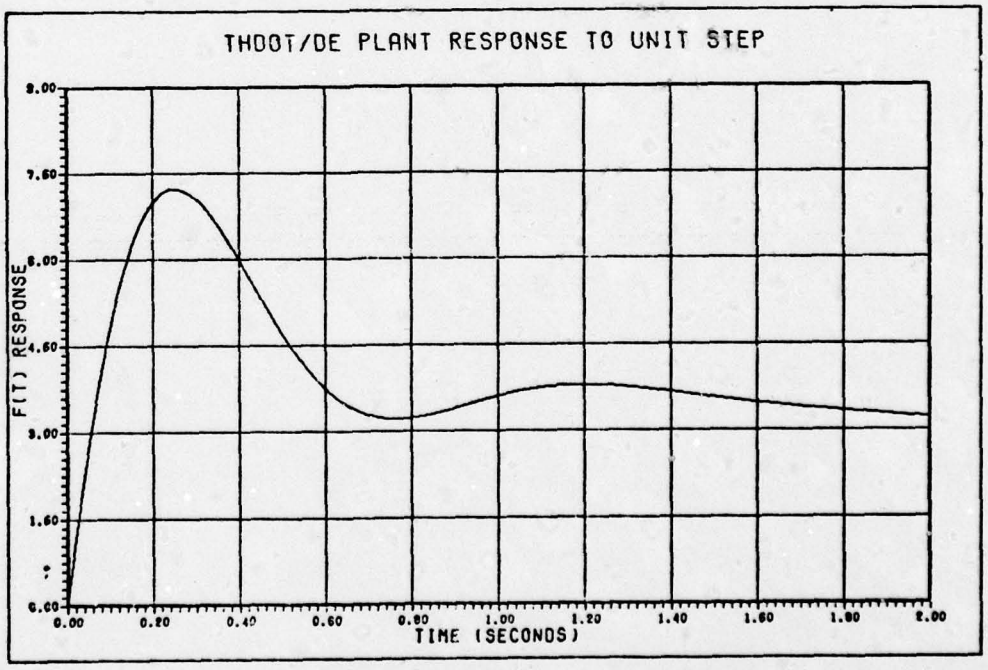


Figure A-14b: $-\dot{\theta}/\delta_e$ Time Response (Phugoid)

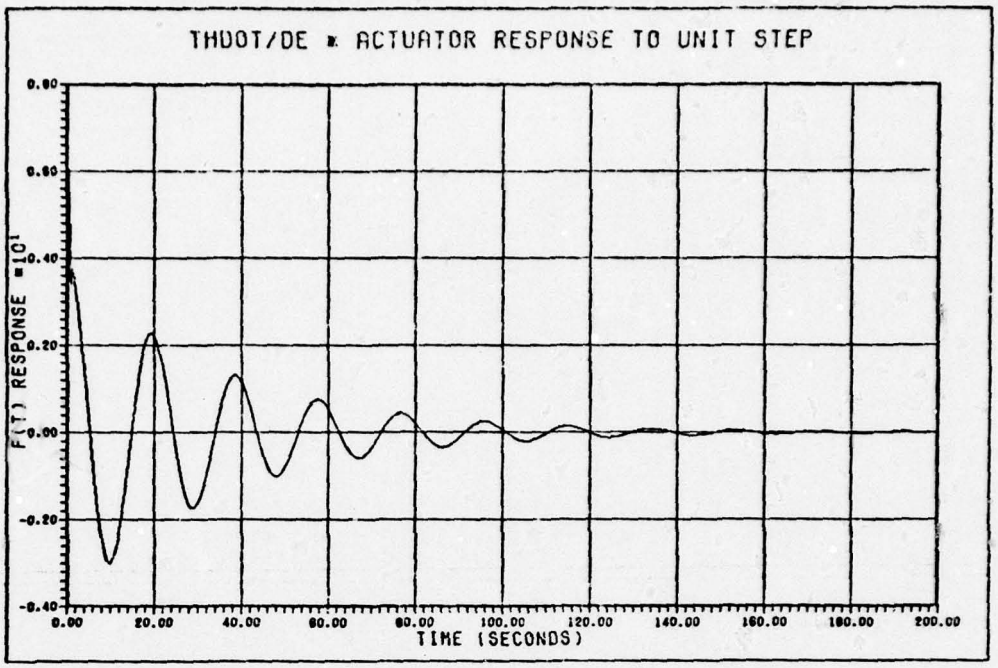


Figure A-15a: $\dot{\theta}/\delta_e$ * Actuator Time Response

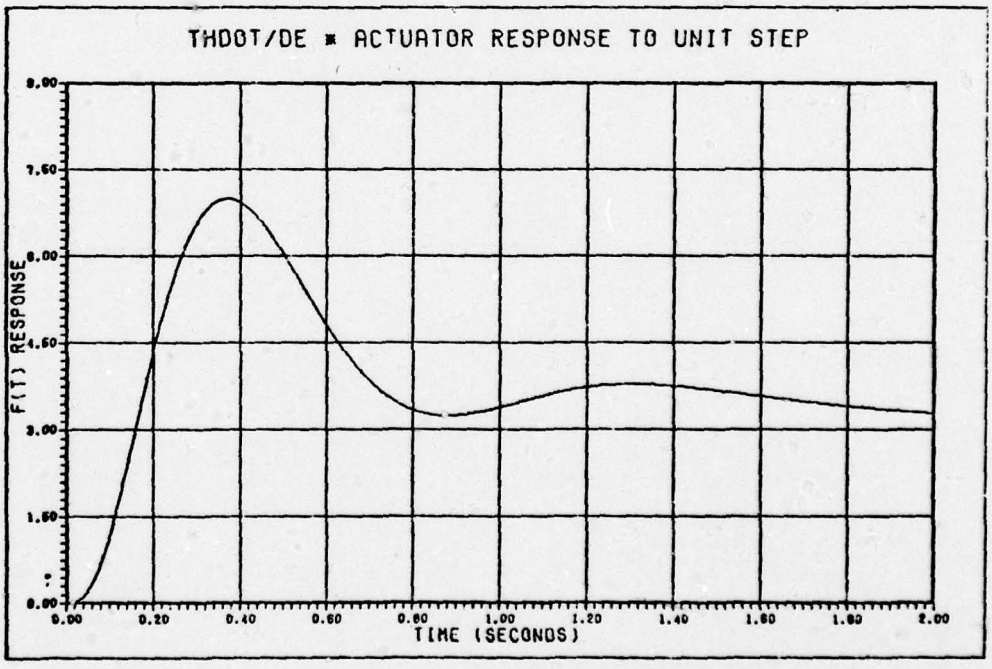


Figure A-15b: $\dot{\theta}/\delta_e$ * Actuator Time Response

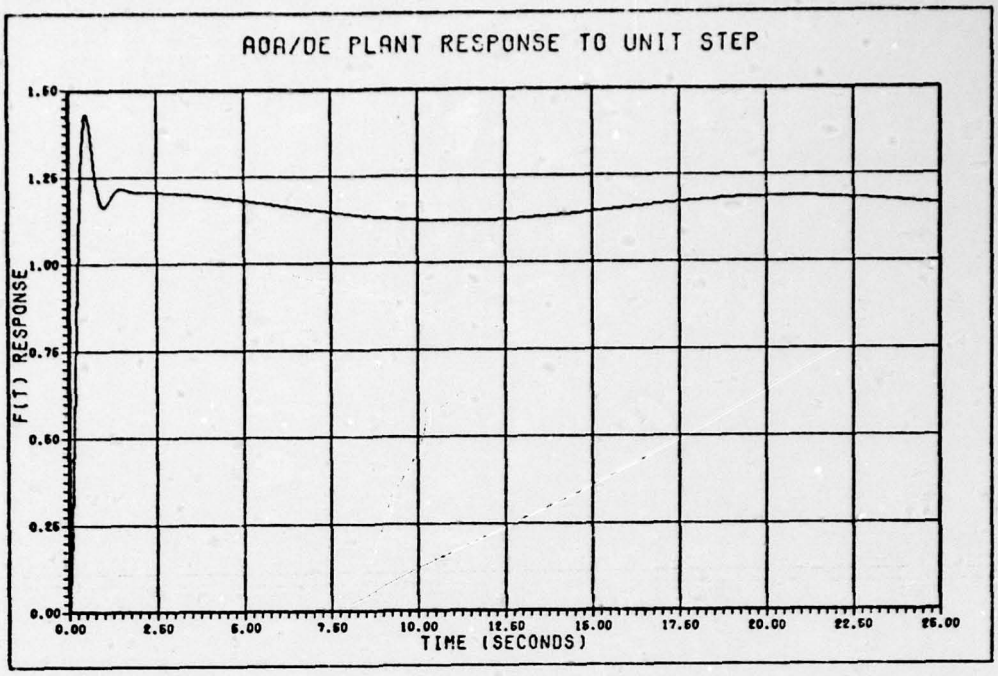


Figure A-16a: $-\alpha/\delta_e$ Time Response

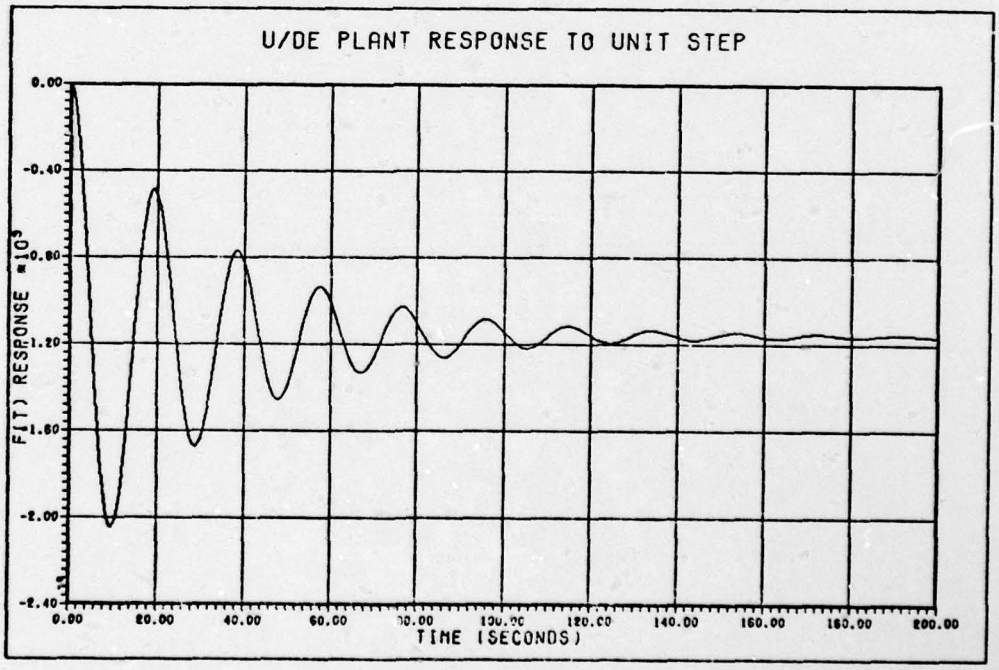


Figure A-16b: $-U/\delta_e$ Time Response (rad/sec)

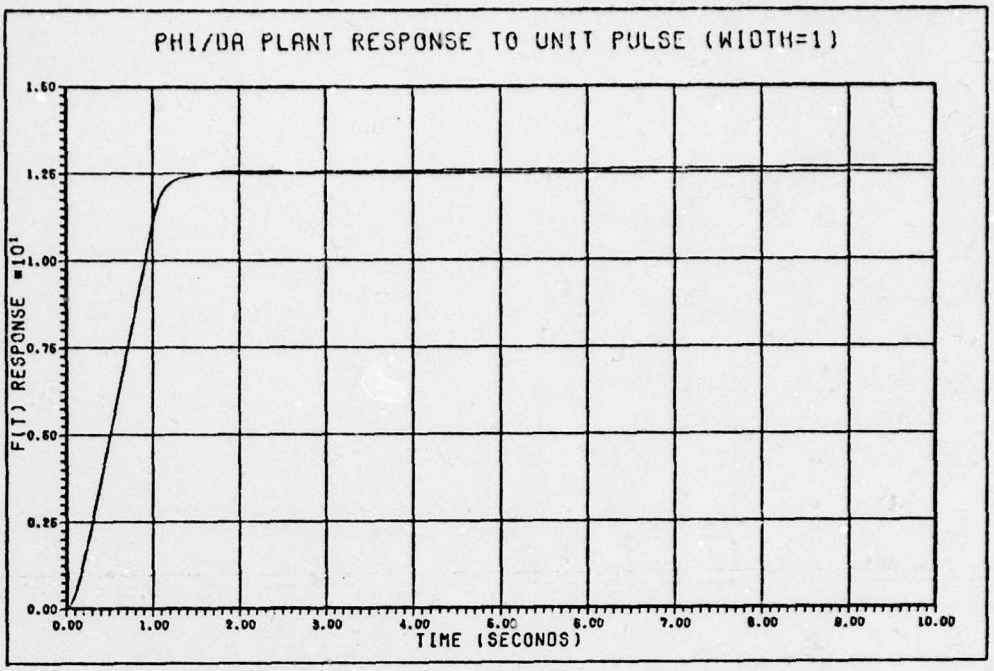


Figure A-17a: ϕ/δ_a Time Response

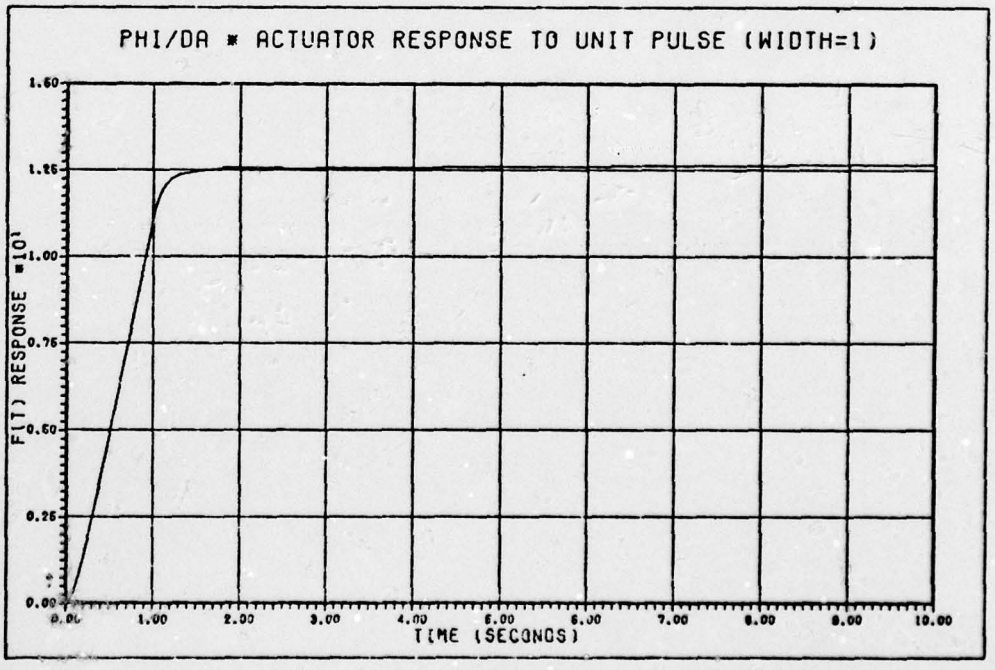


Figure A-17b: ϕ/δ_a * Actuator Time Response

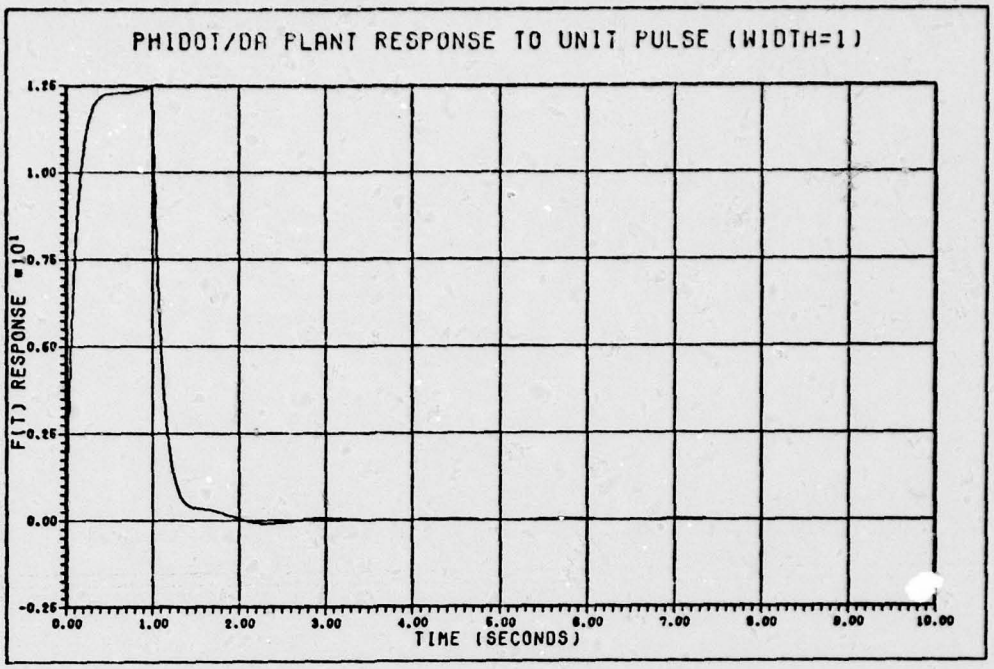


Figure A-18a: $\dot{\phi}/\delta_a$ Time Response

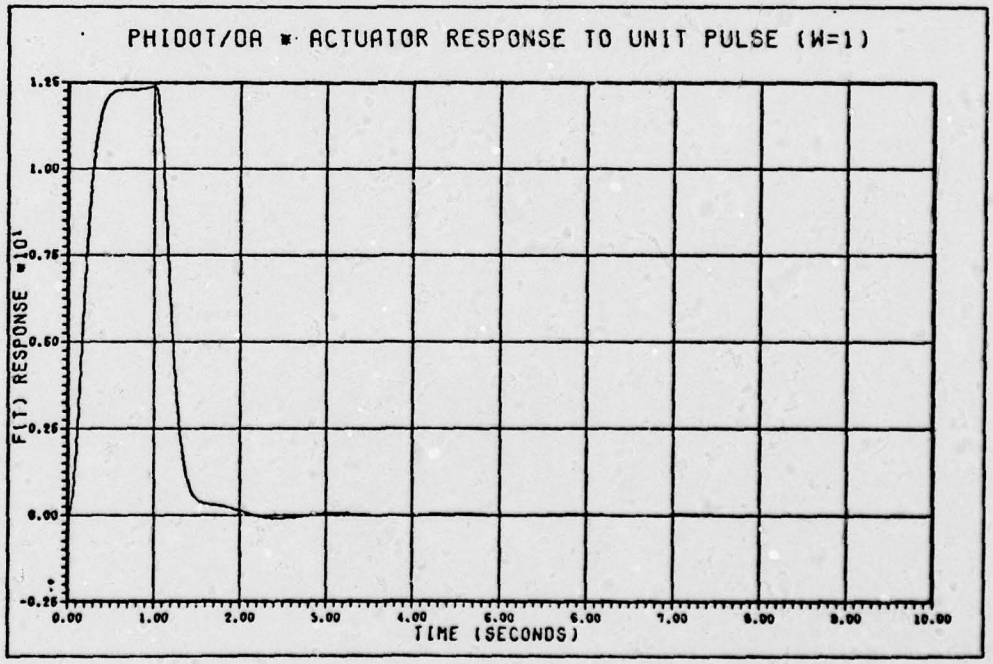


Figure A-18b: $\dot{\phi}/\delta_a$ * Actuator Time Response

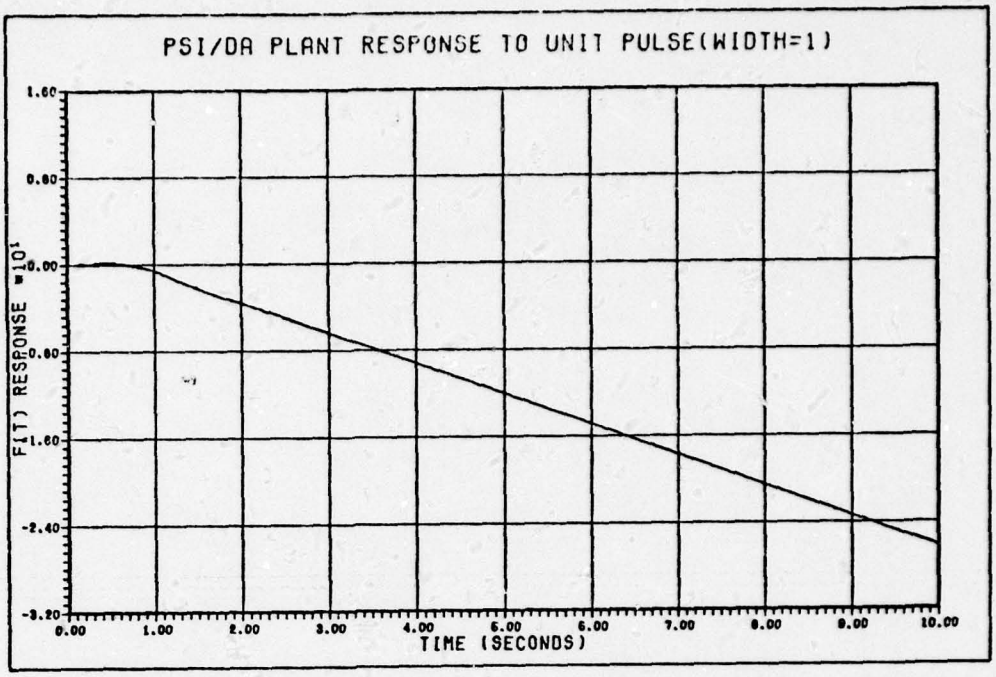


Figure A-19a: ψ/δ_a Time Response

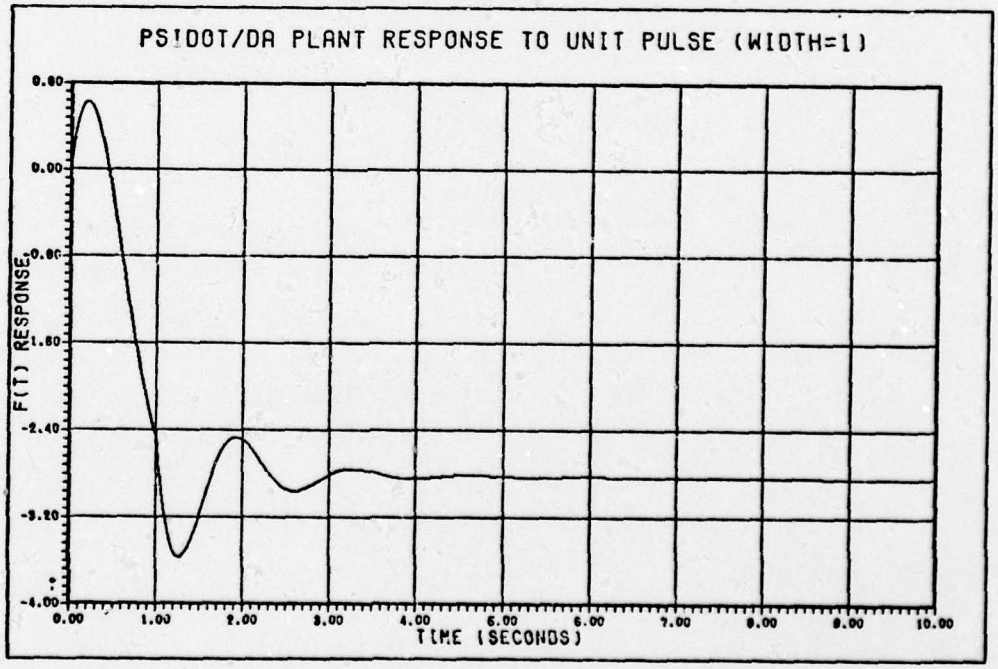


Figure A-19b: $\dot{\psi}/\delta_a$ Time Response

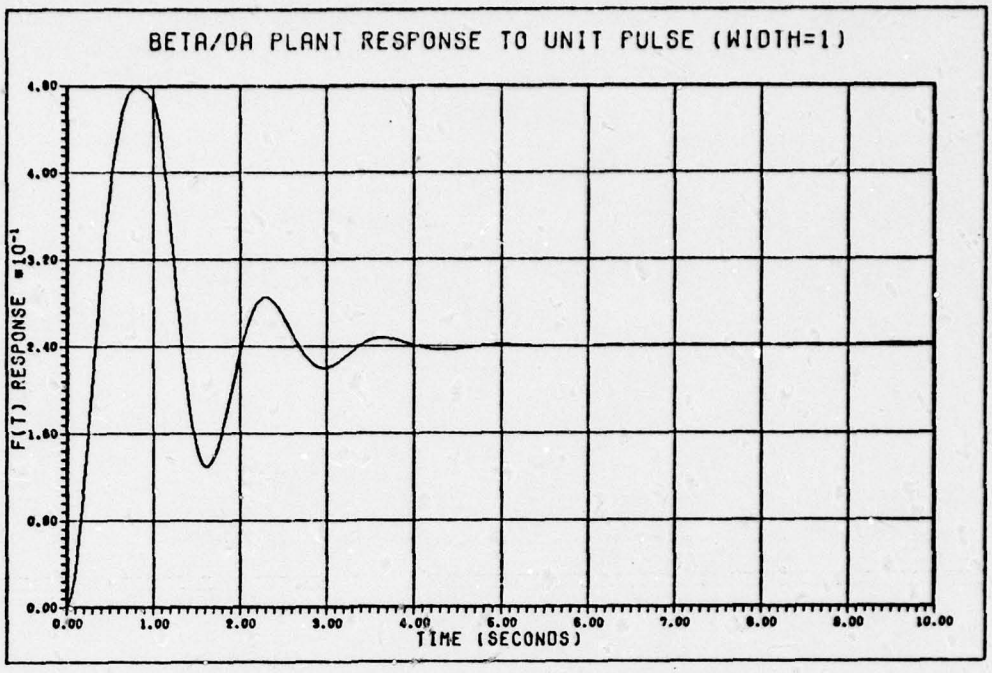


Figure A-20: β/δ_a Time Response

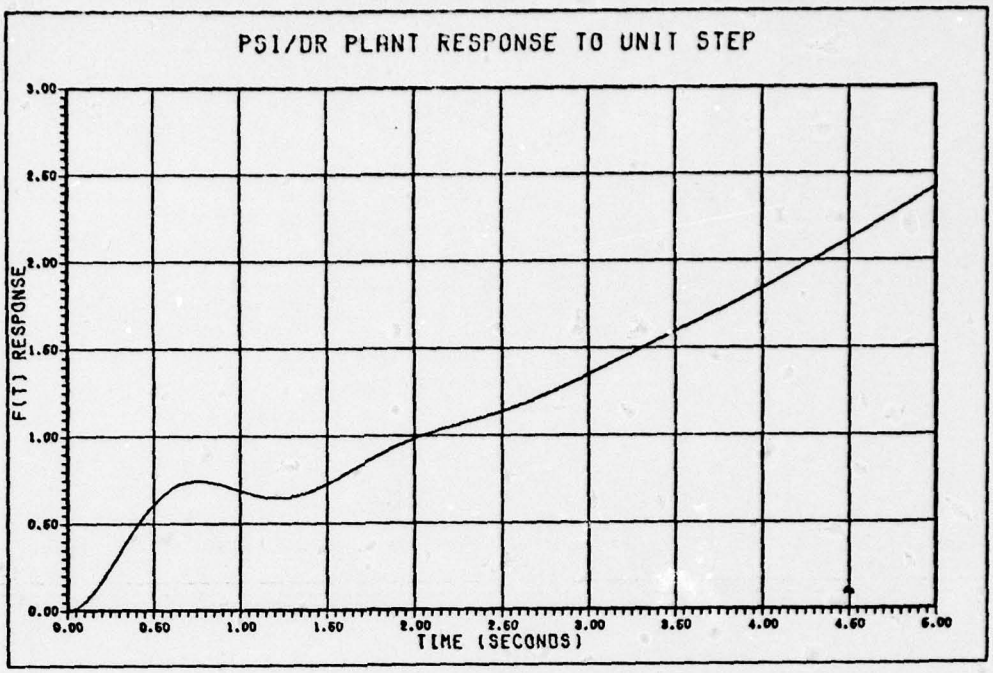


Figure A-21a: ψ/δ_r Time Response (Step)

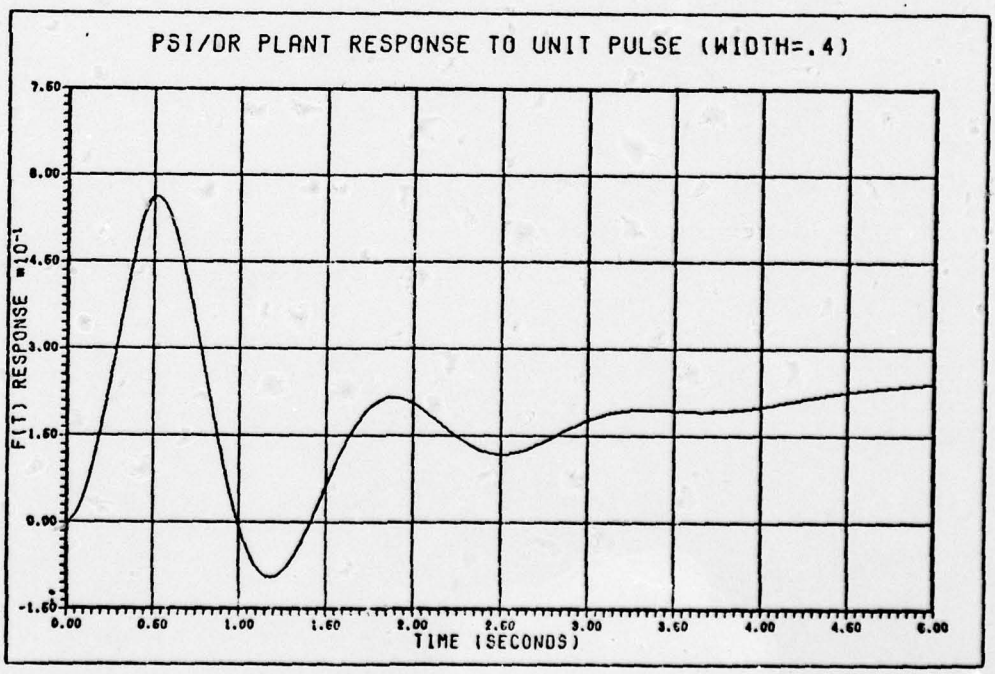


Figure A-21b: ψ/δ_r Time Response (Pulse)

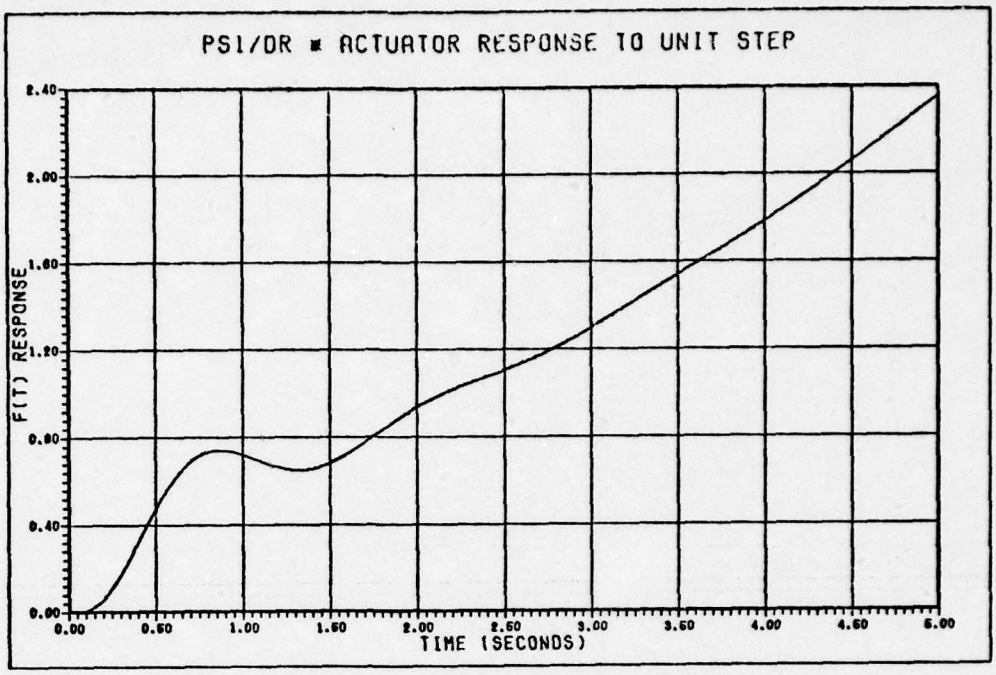


Figure A-22a: ψ/δ_r * Actuator Time Response (Step)

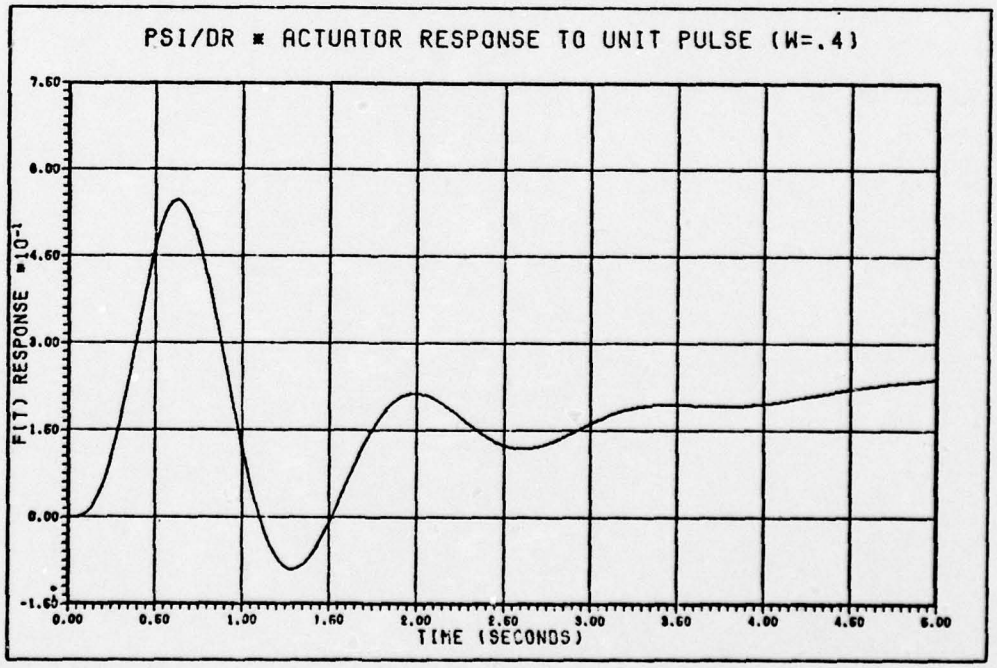


Figure A-22b: ψ/δ_r * Actuator Time Response (Pulse)

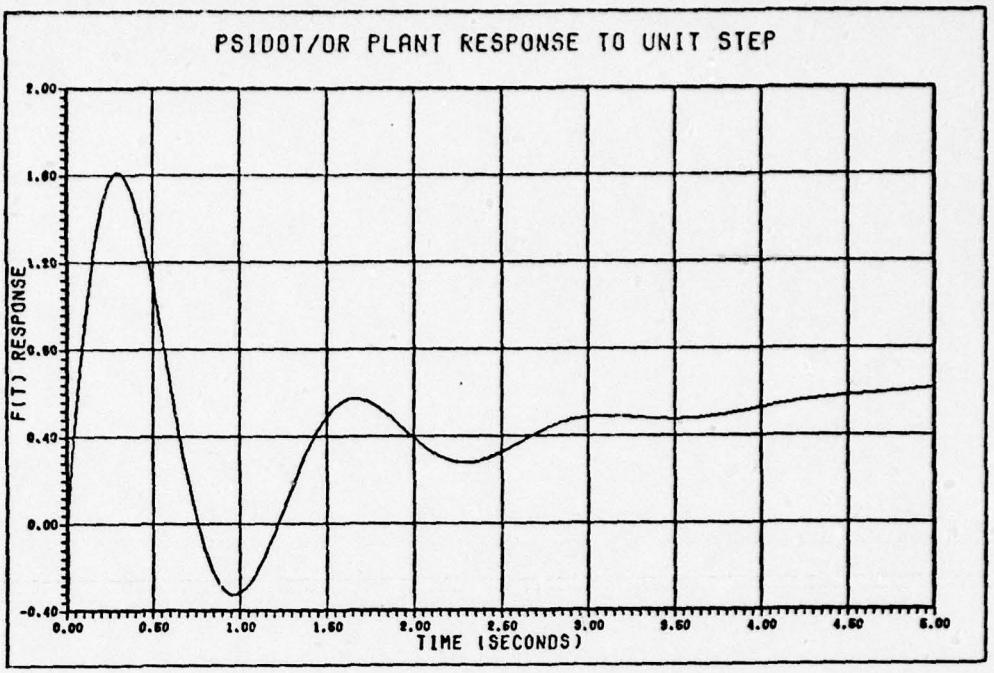


Figure A-23a: $\dot{\psi}/\delta_r$ Time Response (Step)

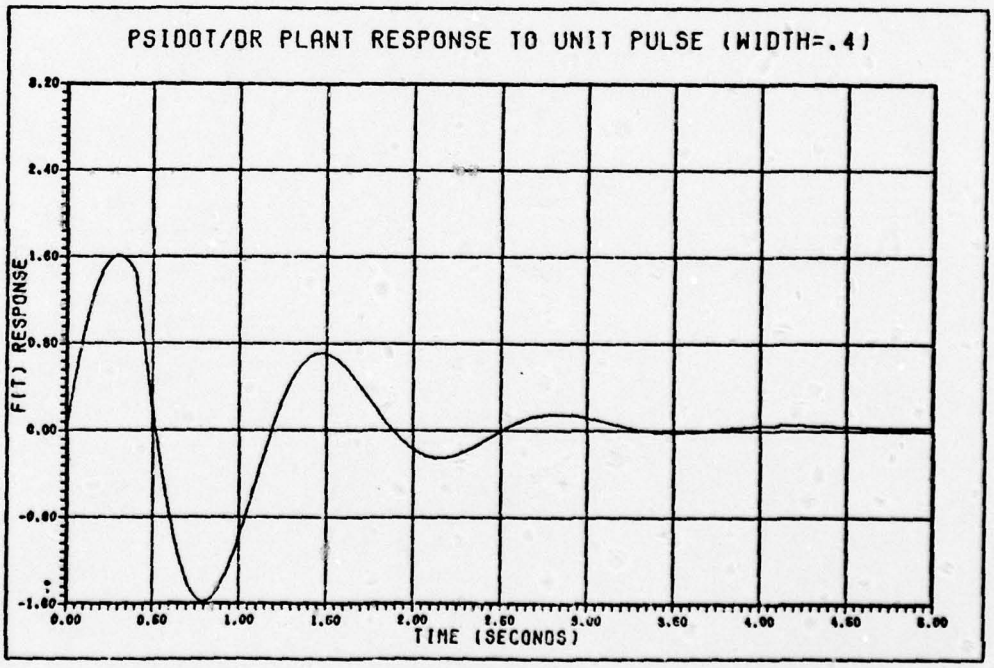


Figure A-23b: $\dot{\psi}/\delta_r$ Time Response (Pulse)

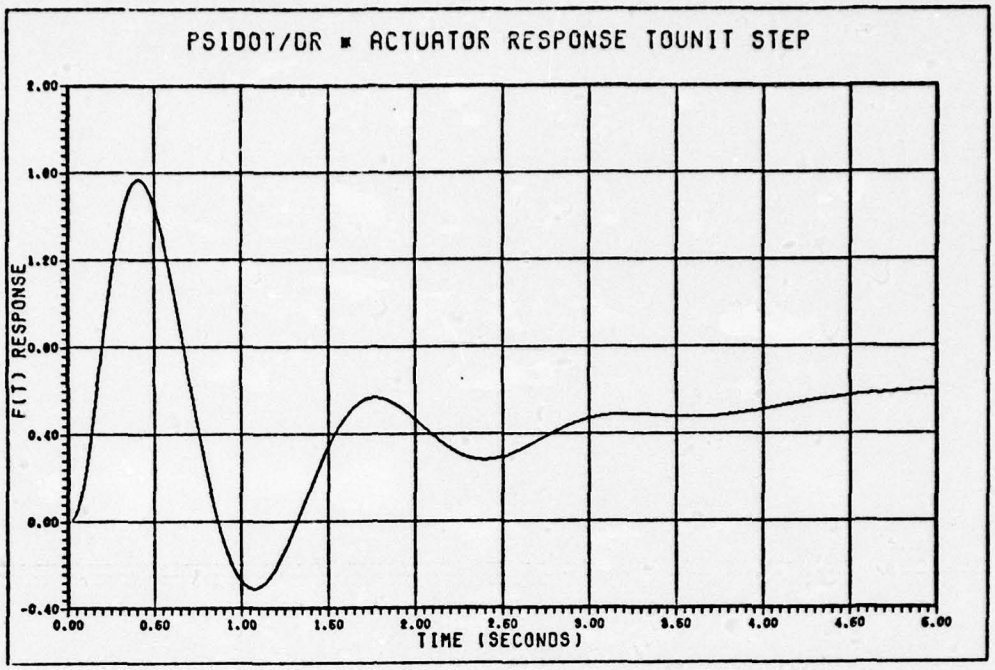


Figure A-24a: $\dot{\psi}/\delta_r$ * Actuator Time Response (Step)

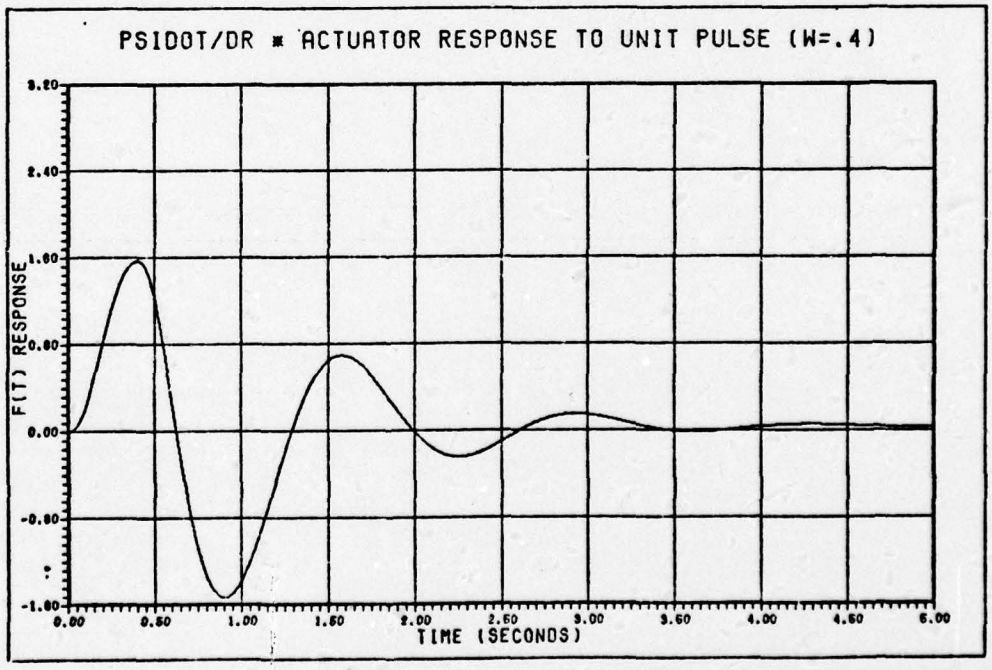


Figure A-24b: $\dot{\psi}/\delta_r$ * Actuator Time Response (Pulse)

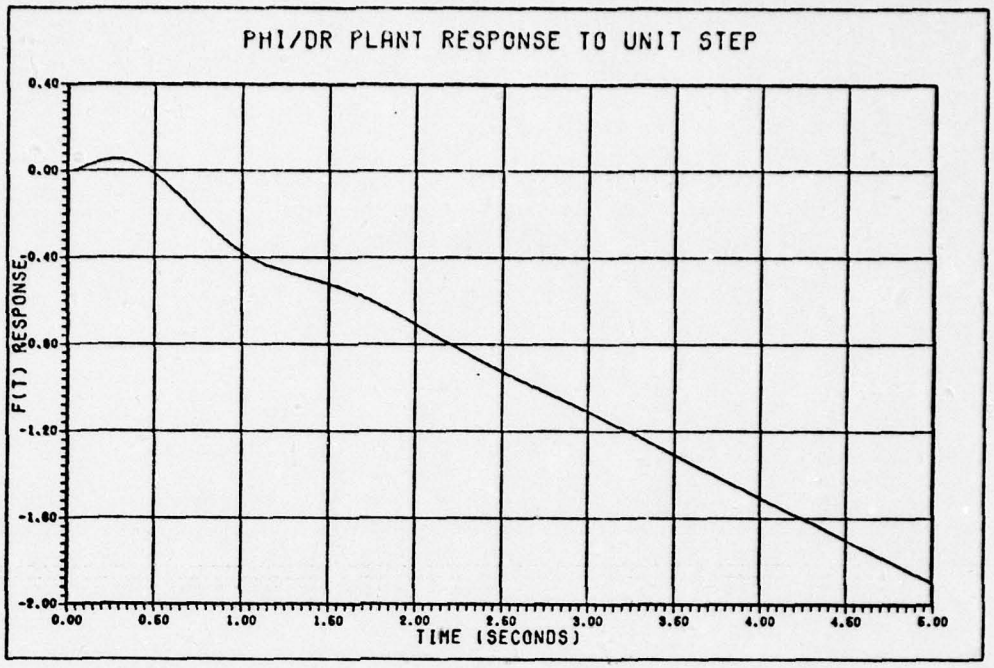


Figure A-25a: ϕ/δ_r Time Response (Step)

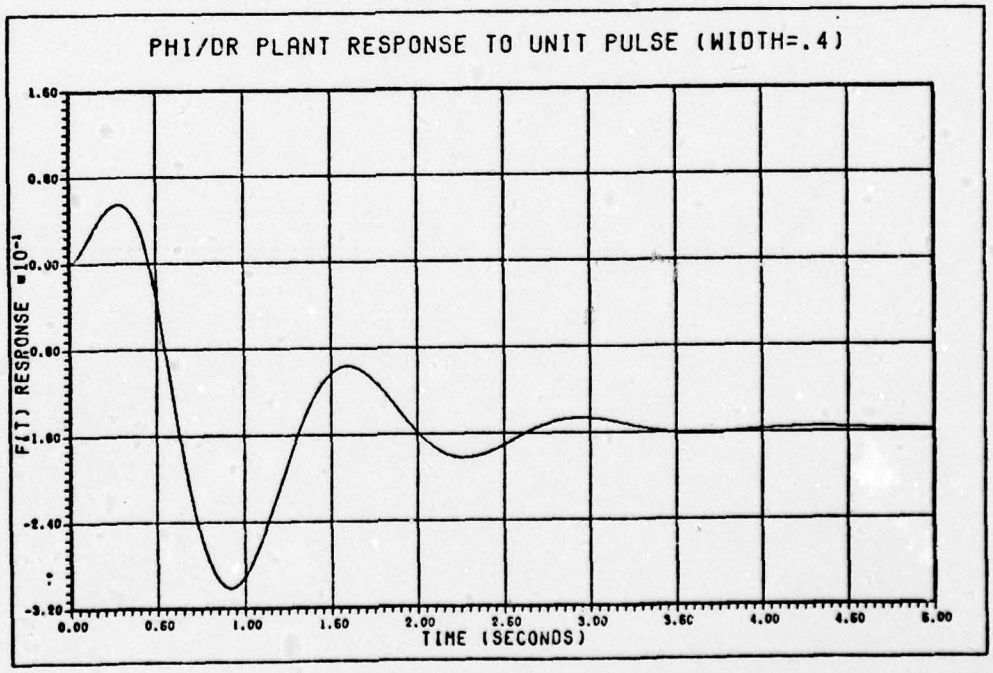


Figure A-25b: ϕ/δ_r Time Response (Pulse)

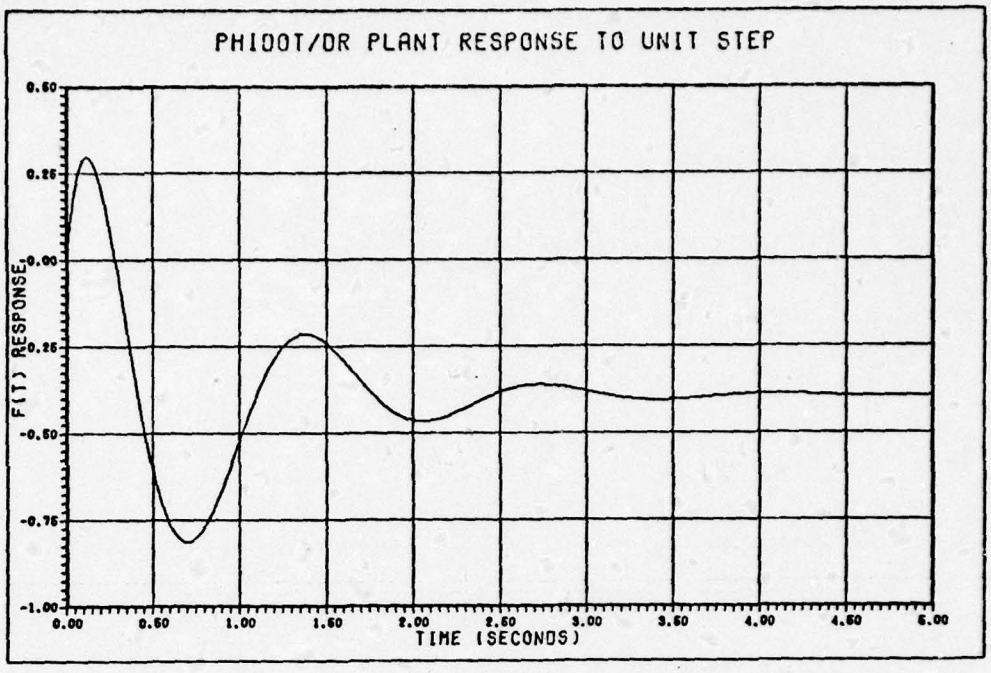


Figure A-26a: $\dot{\phi}/\delta_r$ Time Response (Step)

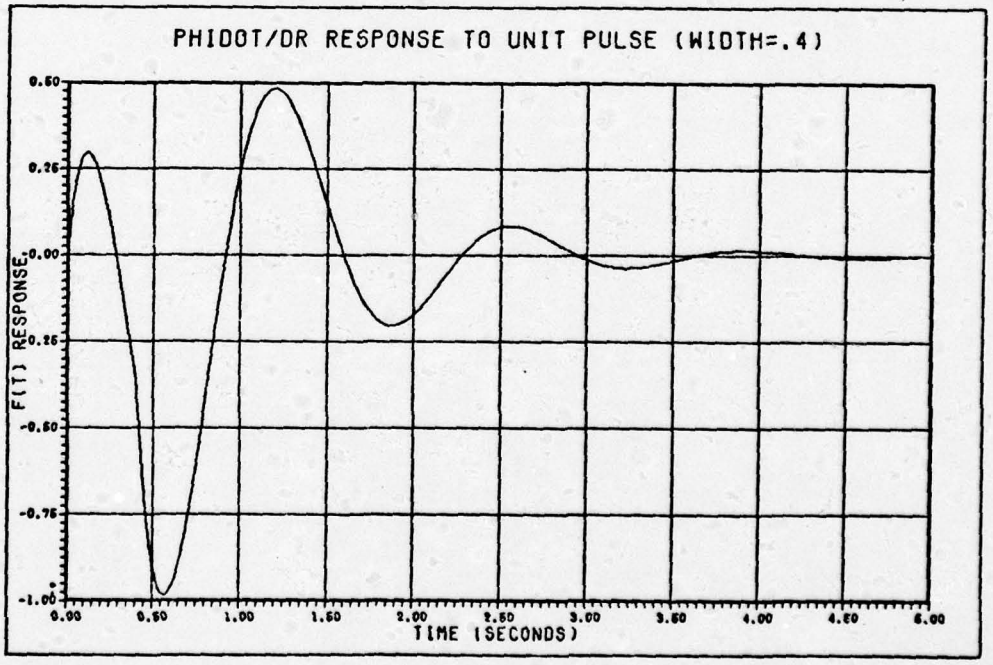


Figure A-26b: $\dot{\phi}/\delta_r$ Time Response (Pulse)

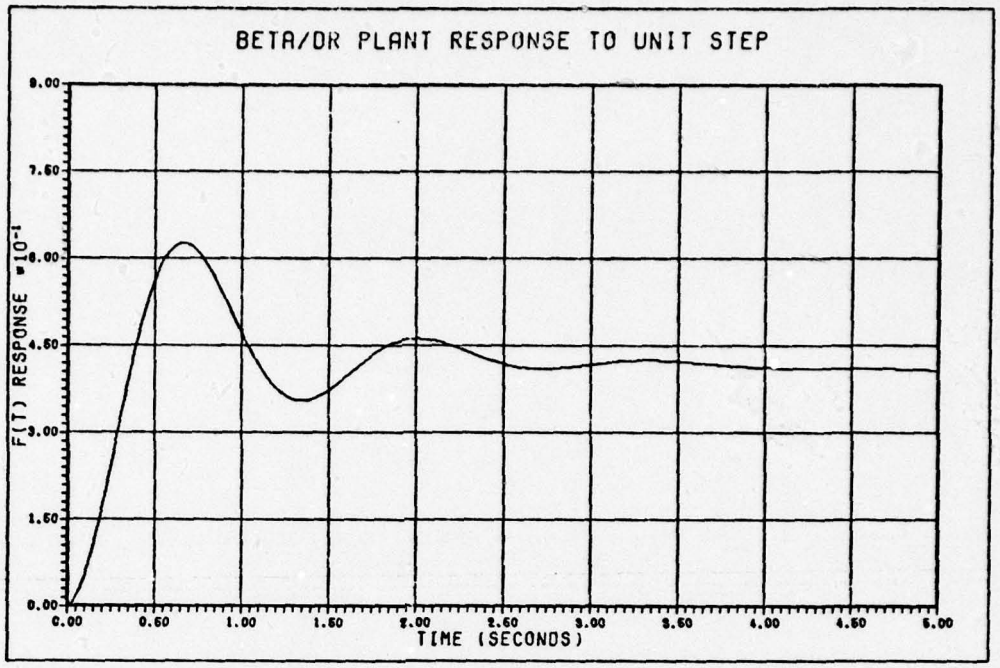


Figure A-27a: β/δ_r Time Response (Step)

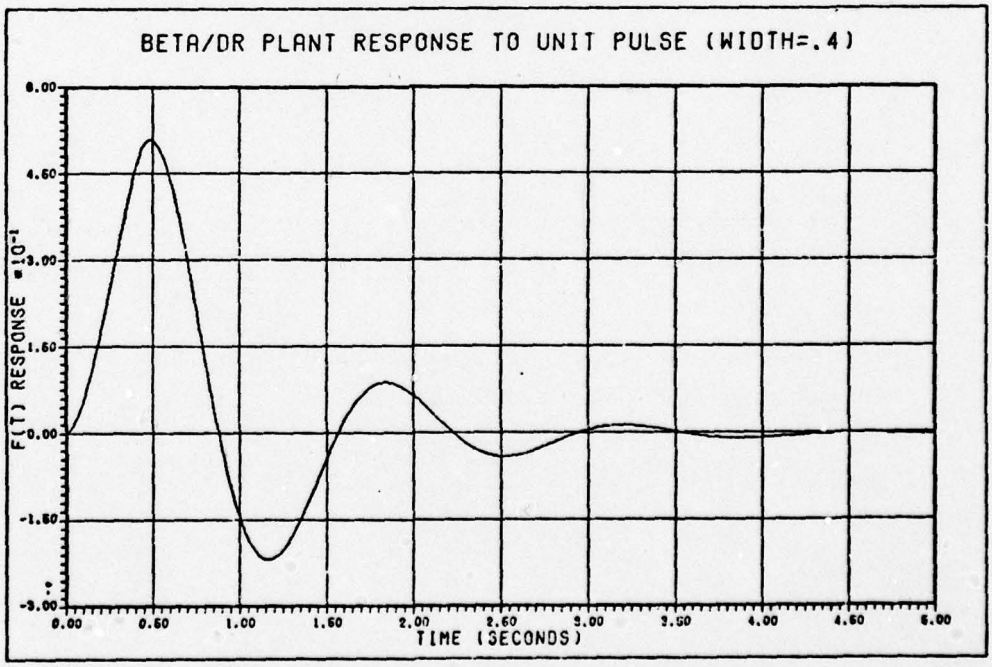


Figure A-27b: β/δ_r Time Response (Pulse)

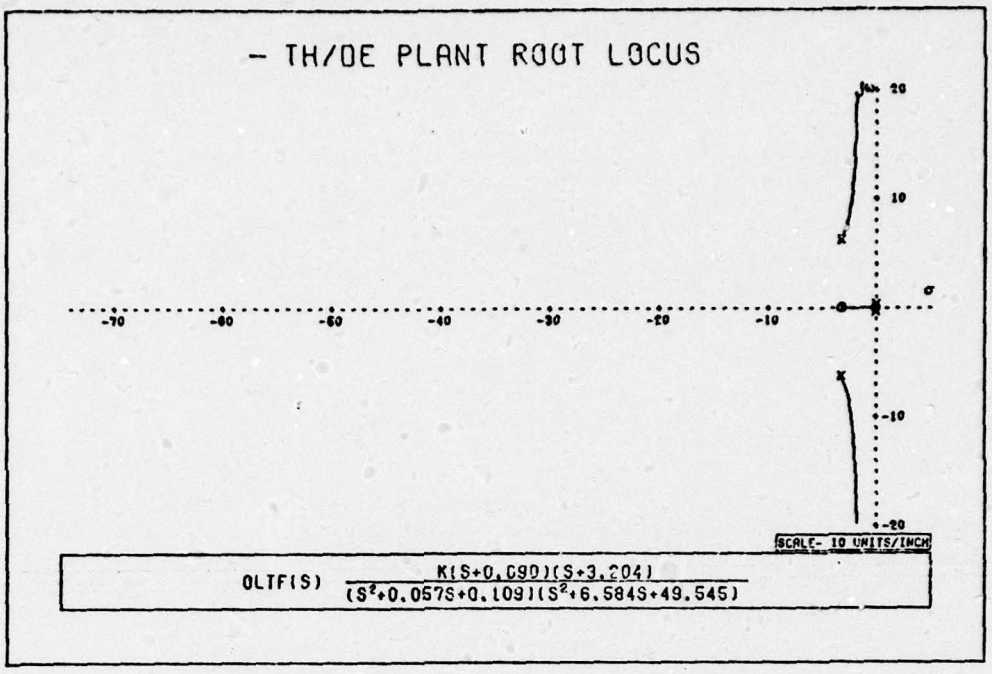


Figure A-28a: $-\theta/\delta_e$ Root Locus

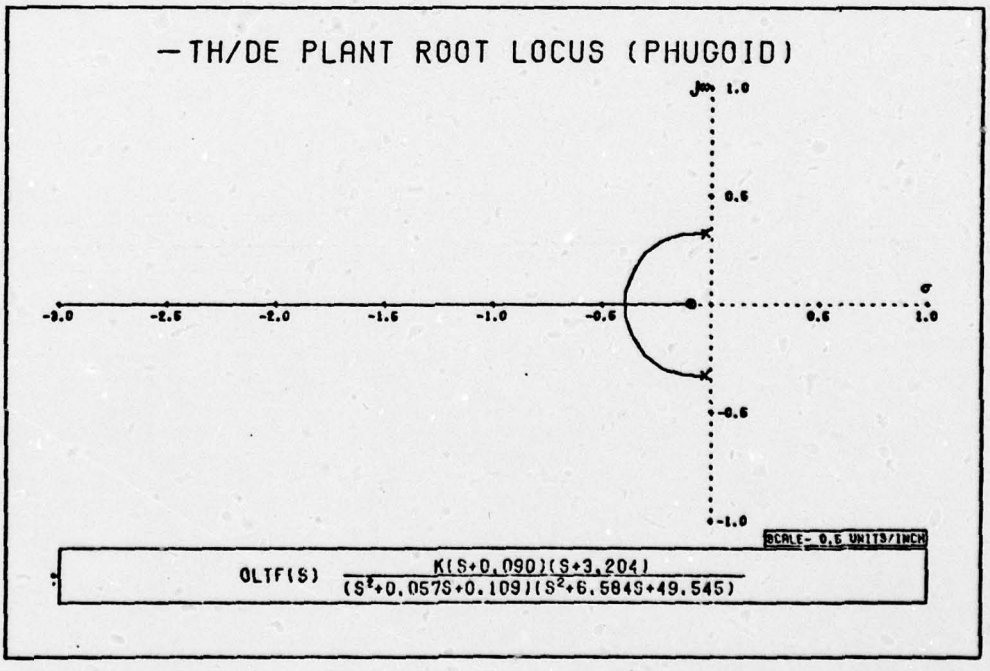


Figure A-28b: $-\theta/\delta_e$ Root Locus (Phugoid)

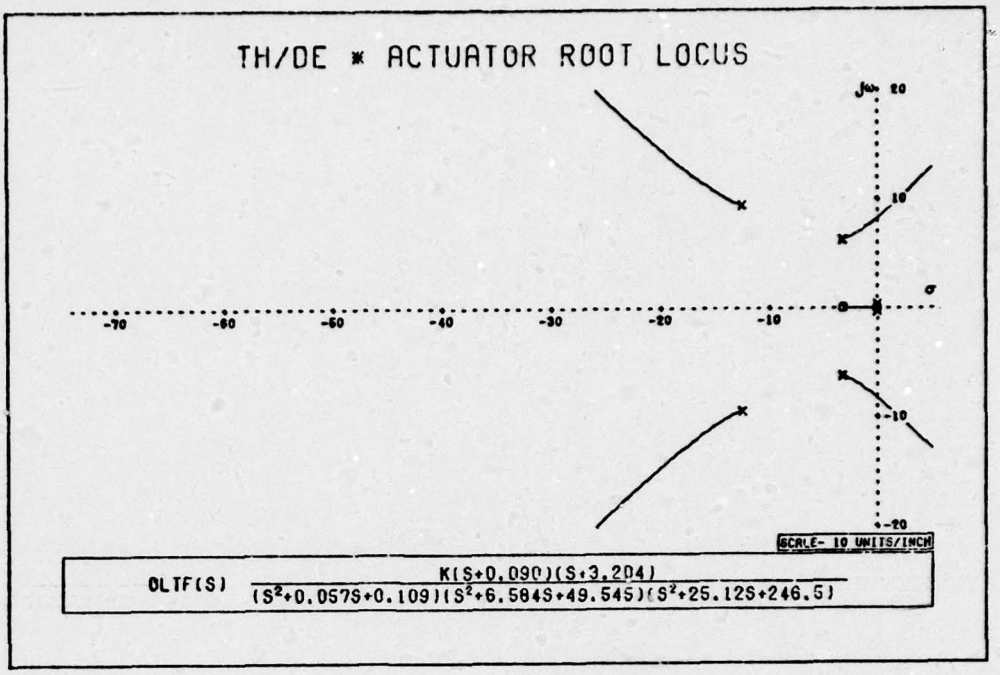


Figure A-29a: θ/δ_e * Actuator Root Locus

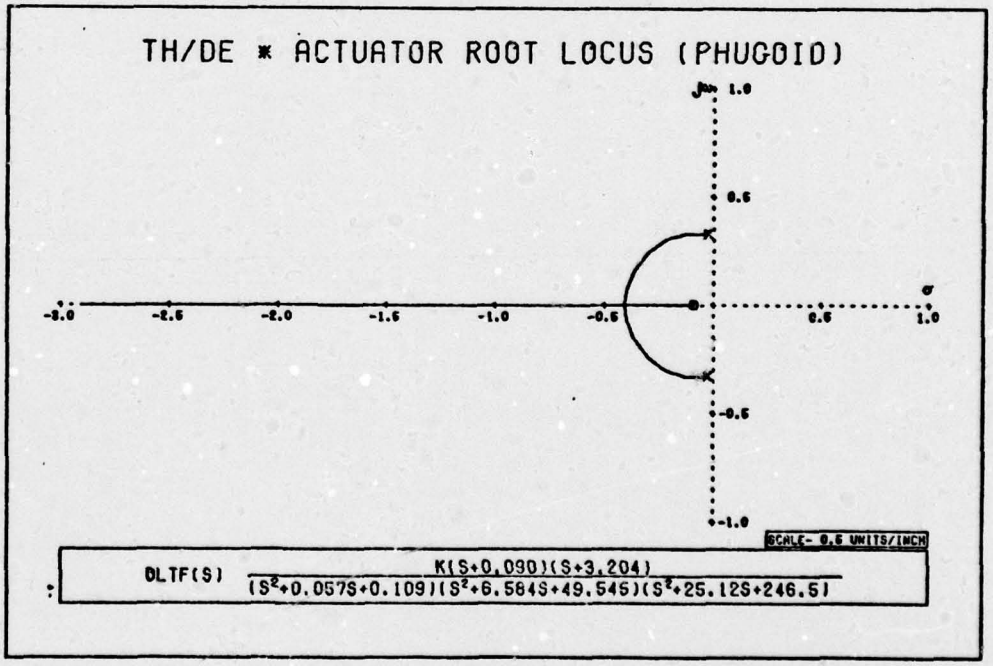


Figure A-29b: θ/δ_e * Actuator Root Locus (Phugoid)

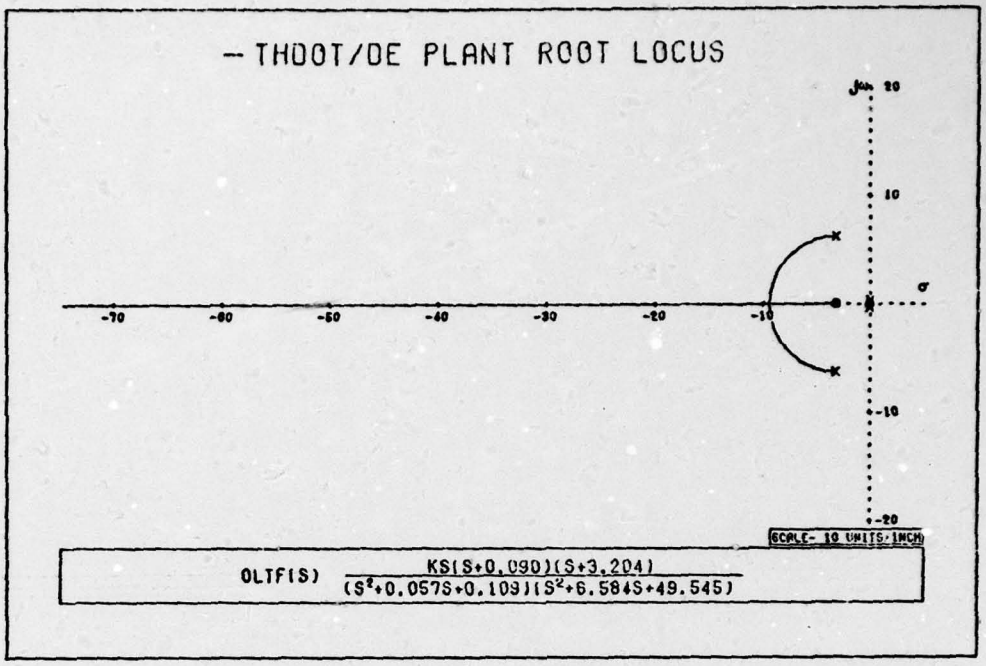


Figure A-30a: $-\dot{\theta}/\delta_e$ Root Locus

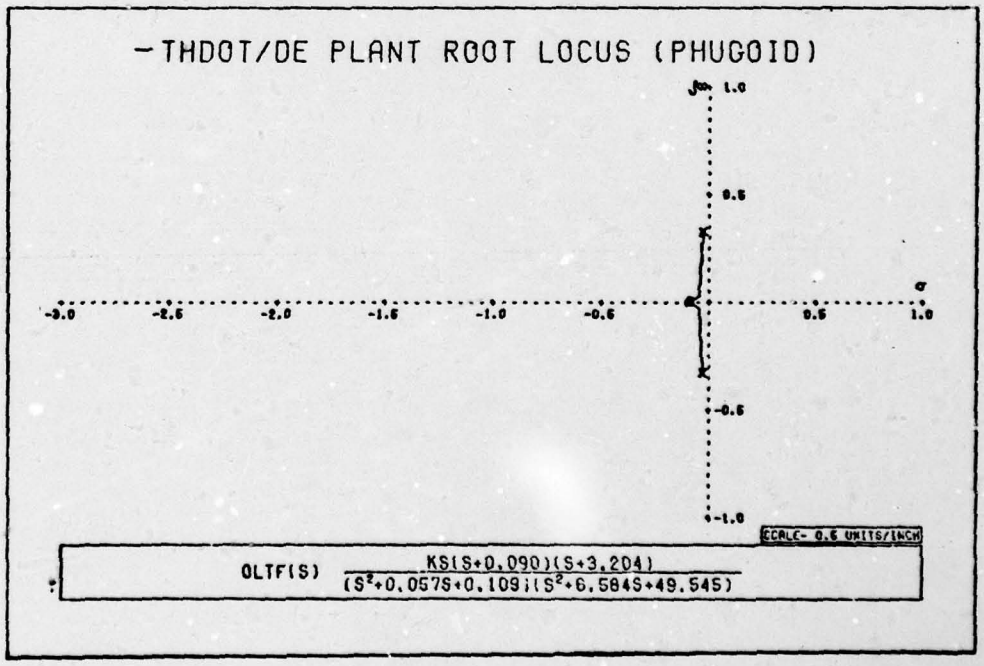


Figure A-30b: $-\dot{\theta}/\delta_e$ Root Locus (Phugoid)

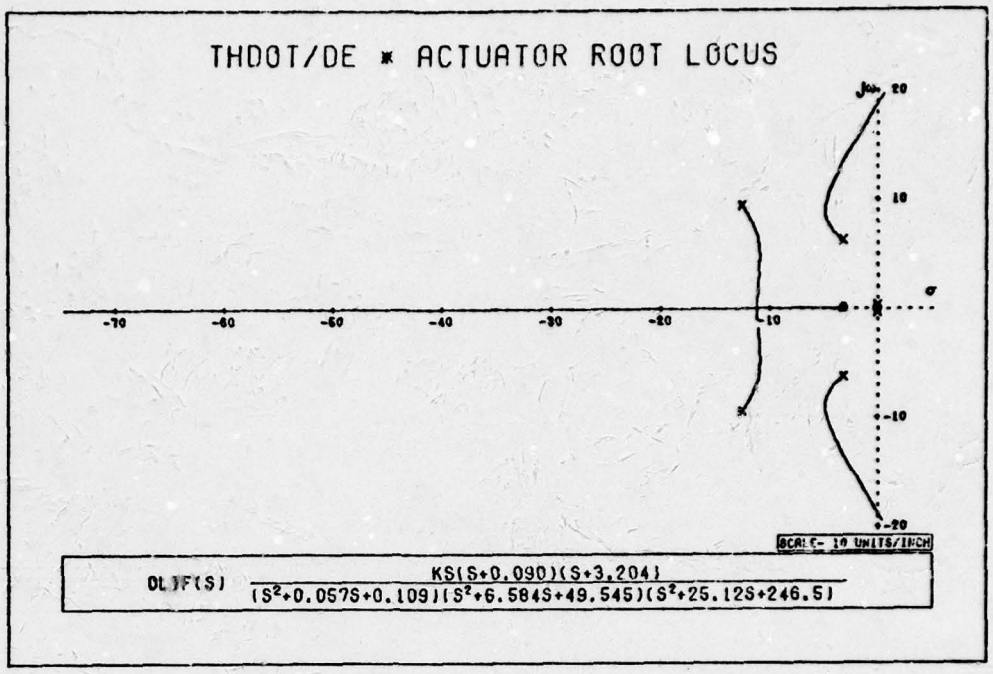


Figure A-31a: $\dot{\theta}/\delta_e$ * Actuator Root Locus

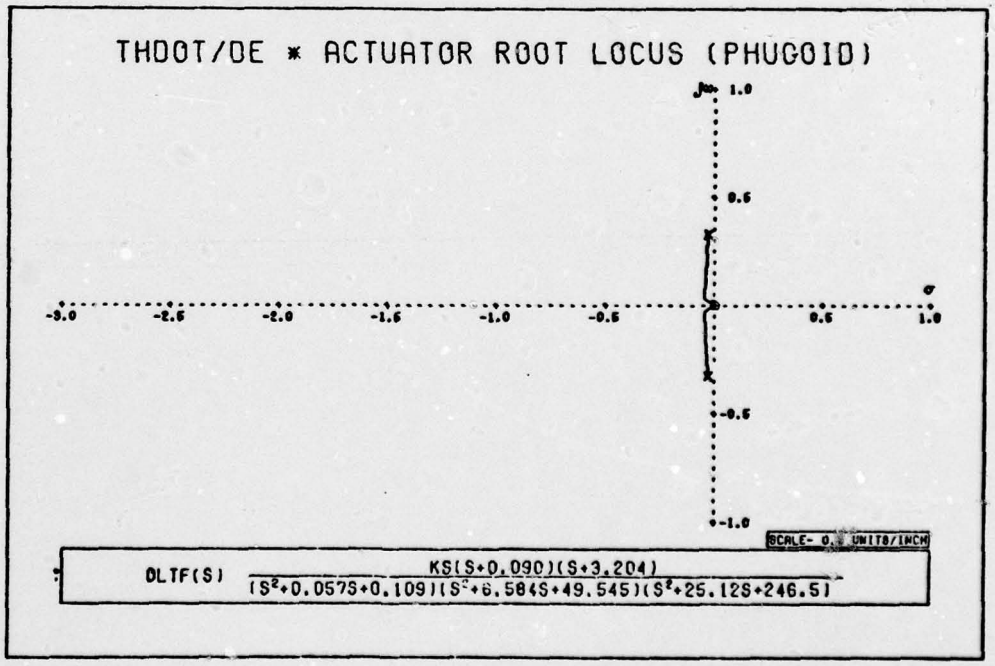


Figure A-31b: $\dot{\theta}/\delta_e$ * Actuator Root Locus (Phugoid)

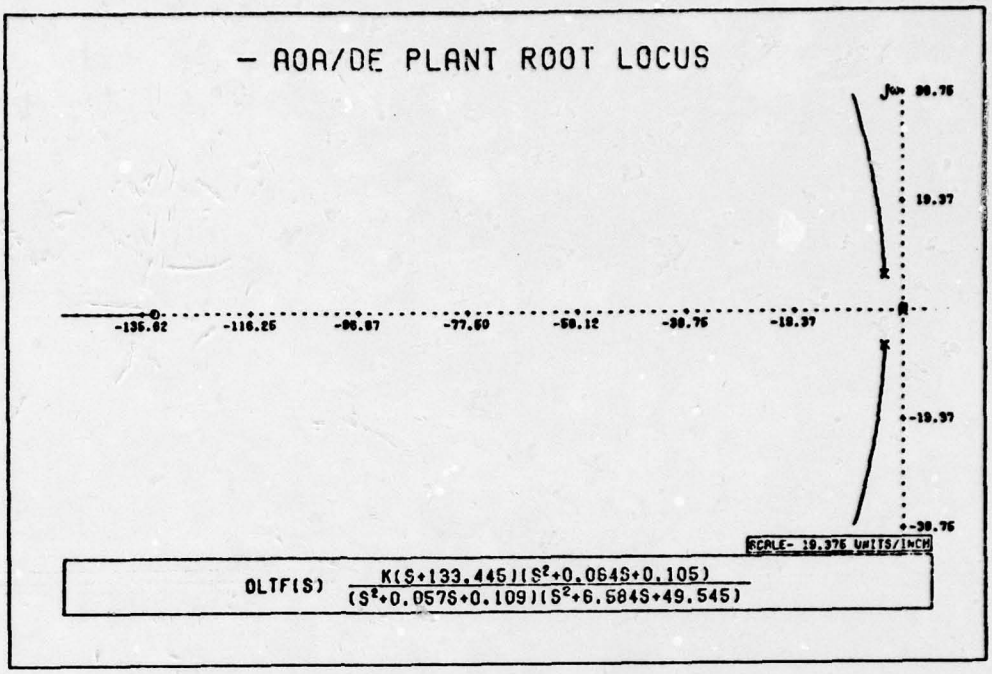


Figure A-32a: $-\alpha/\delta_e$ Root Locus

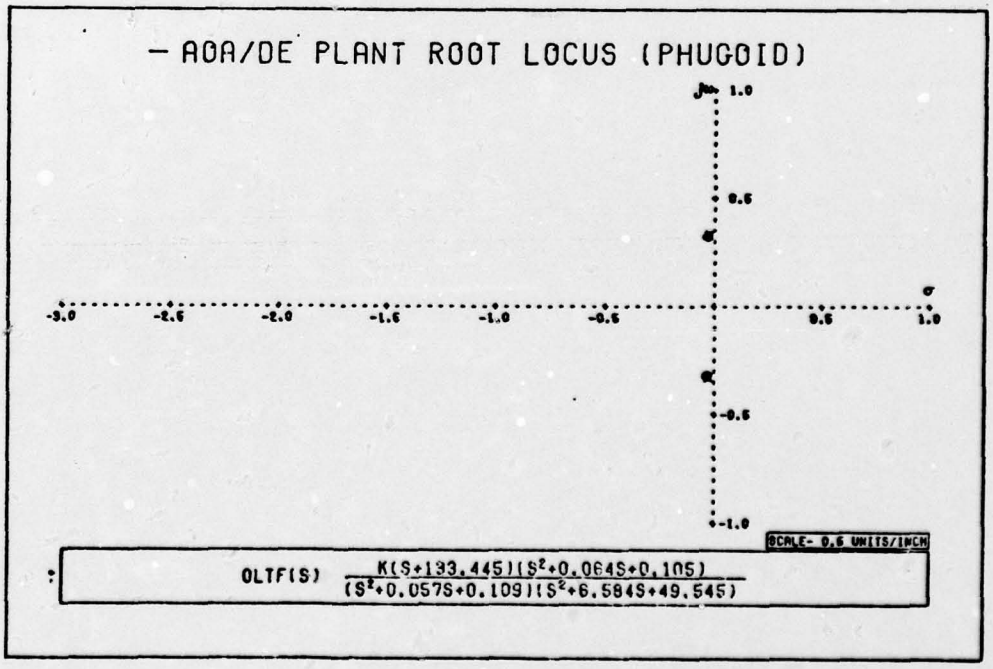


Figure A-32b: $-\alpha/\delta_e$ Root Locus (Phugoid)

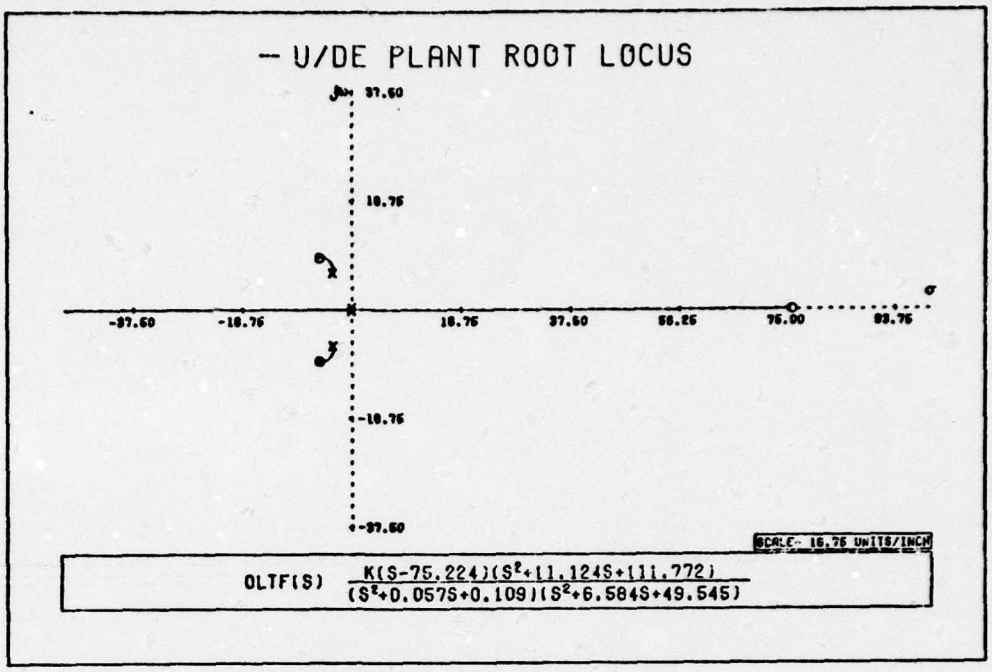


Figure A-33a: $-U/\delta_e$ Root Locus

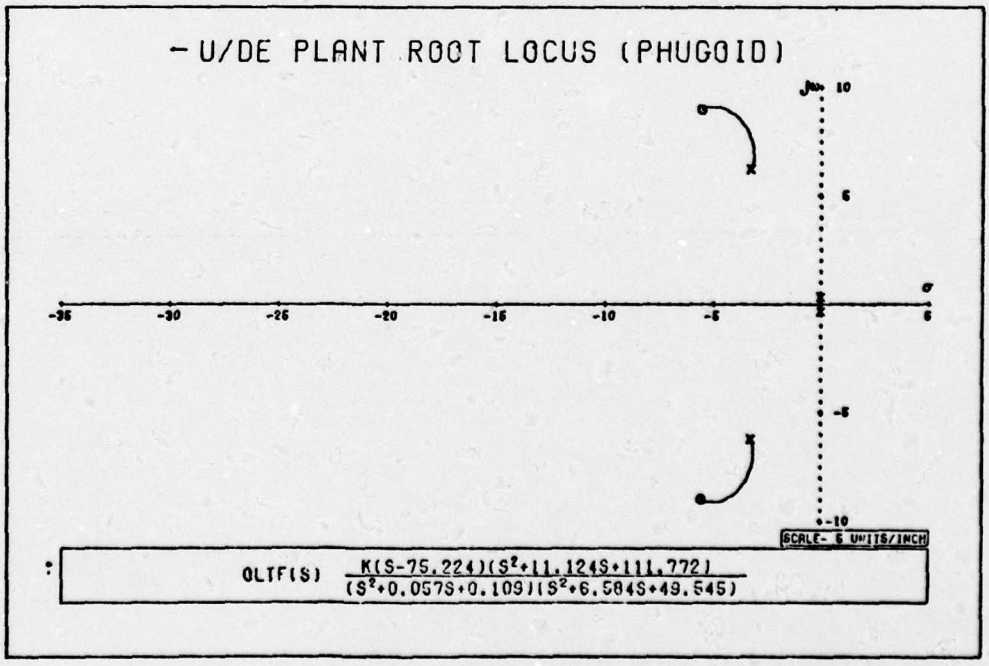


Figure A-33b: $-U/\delta_e$ Root Locus

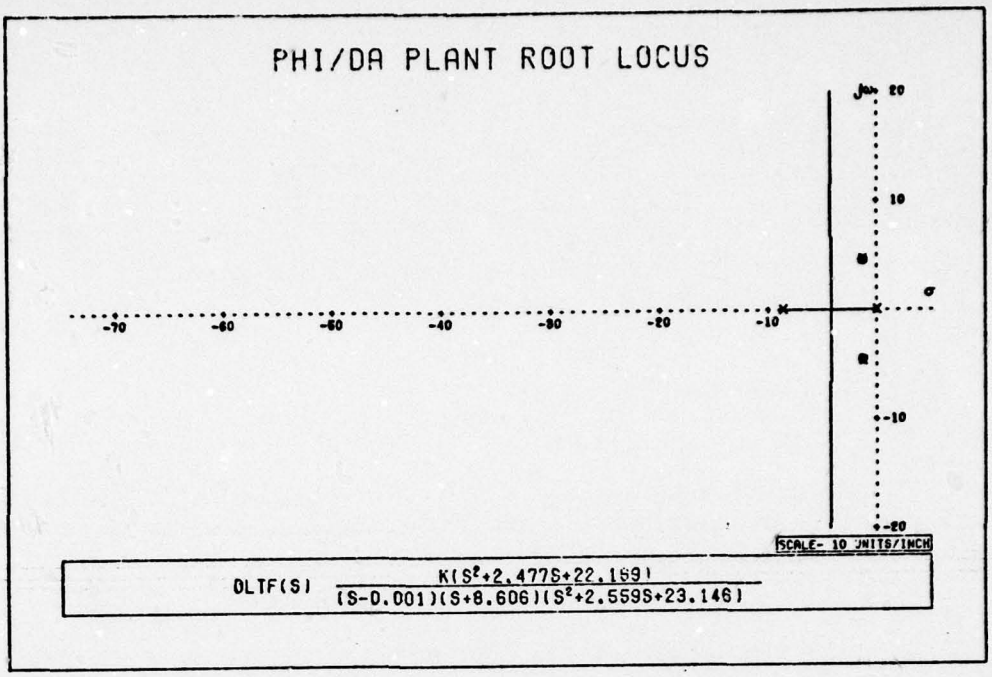


Figure A-34a: ϕ/δ_a Root Locus

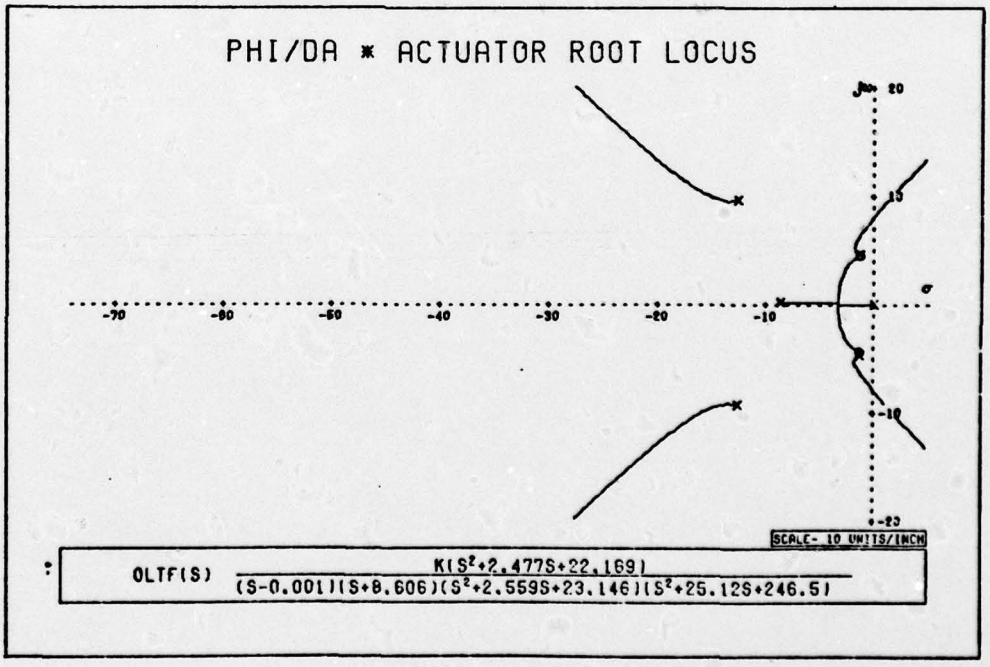


Figure A-34b: ϕ/δ_a * Actuator Root Locus

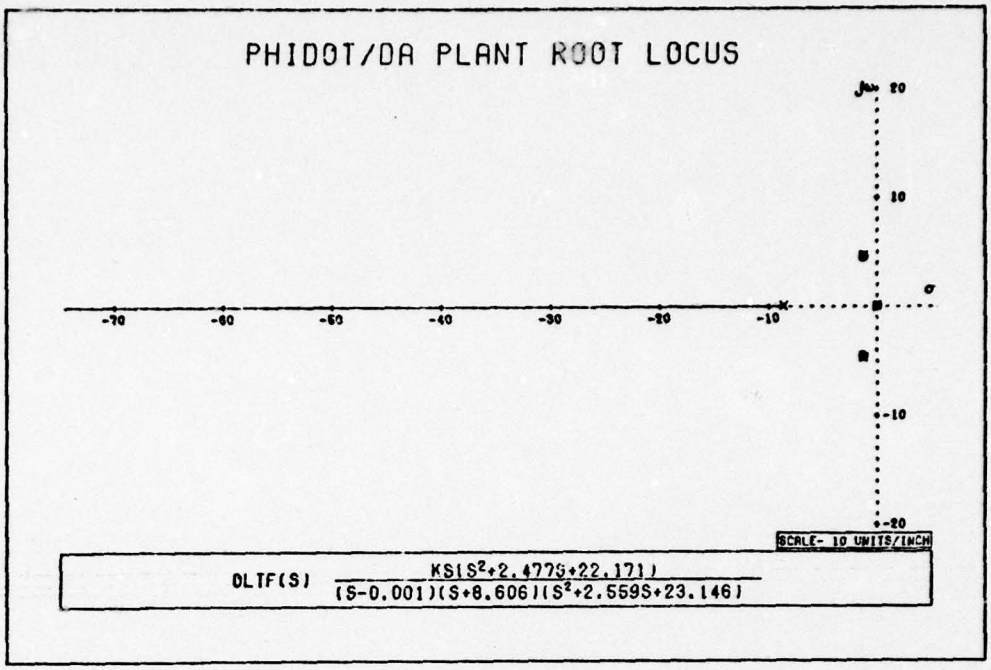


Figure A-35a: $\dot{\phi}/\delta_a$ Root Locus

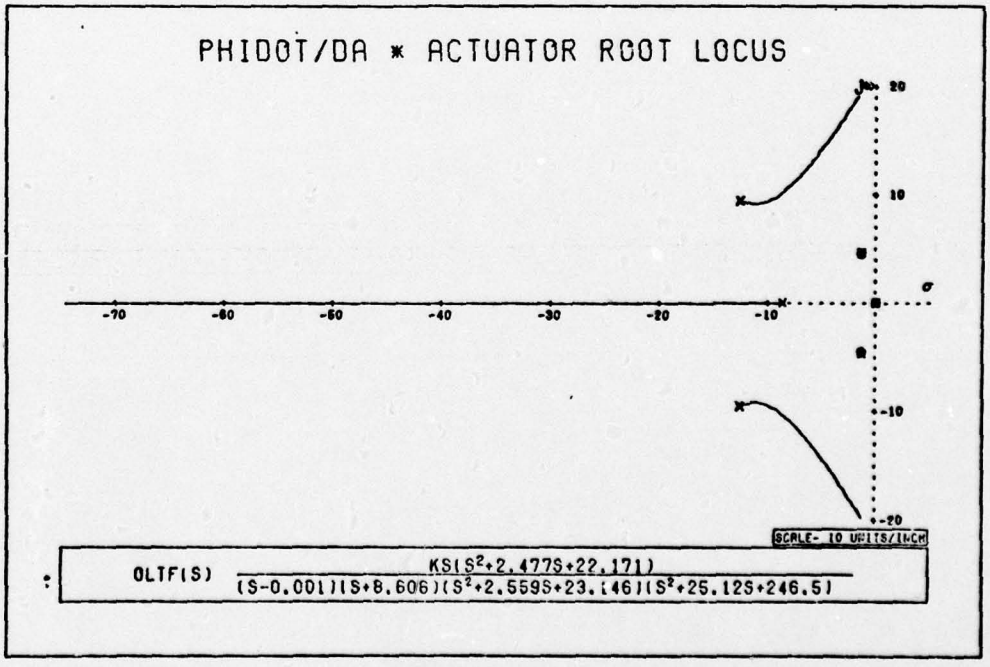


Figure A-35b: $\dot{\phi}/\delta_a$ * Actuator Root Locus

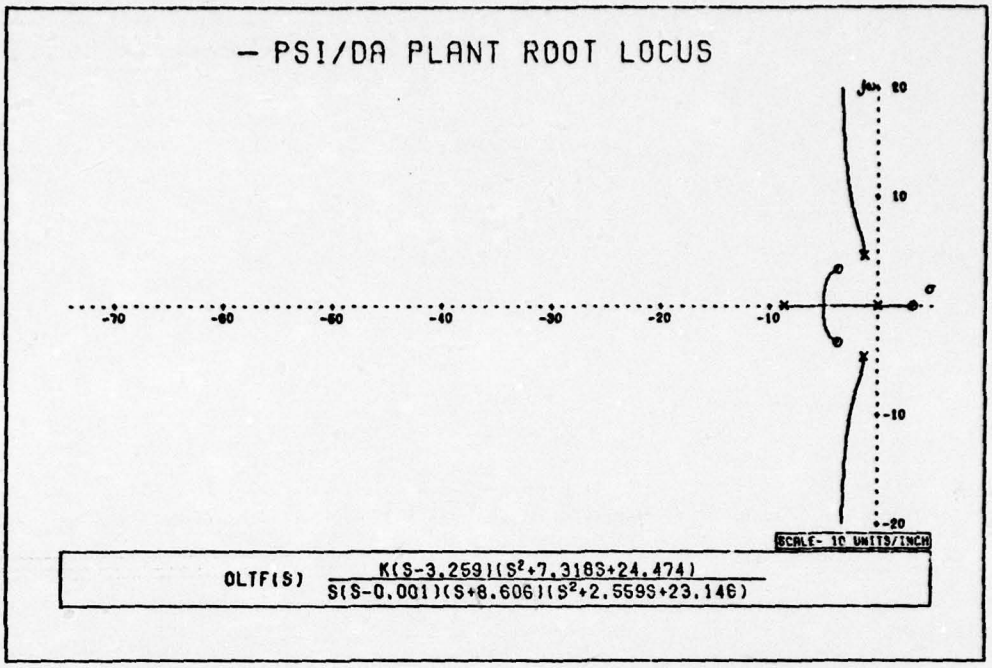


Figure A-36a: $-\psi/\delta_a$ Root Locus

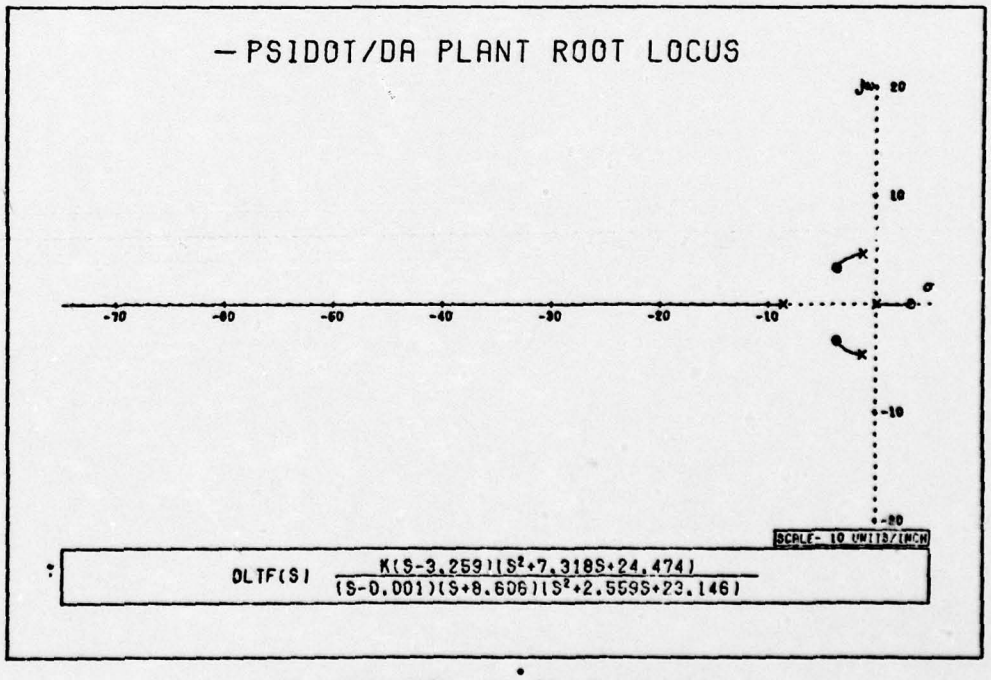


Figure A-36b: $-\dot{\psi}/\delta_a$ Root Locus

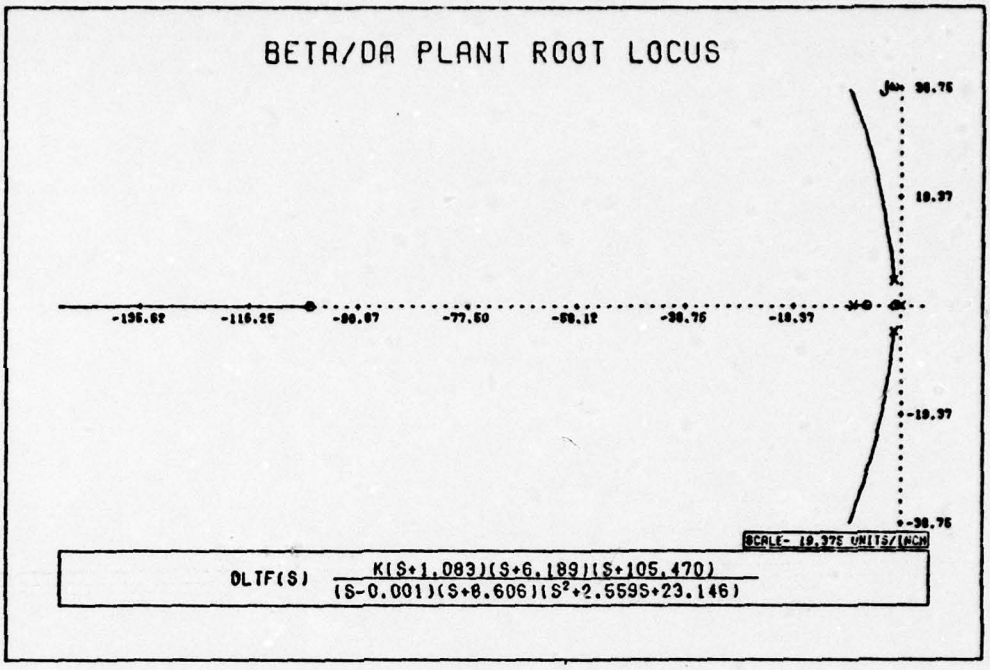


Figure A-37: β/δ_a Root Locus

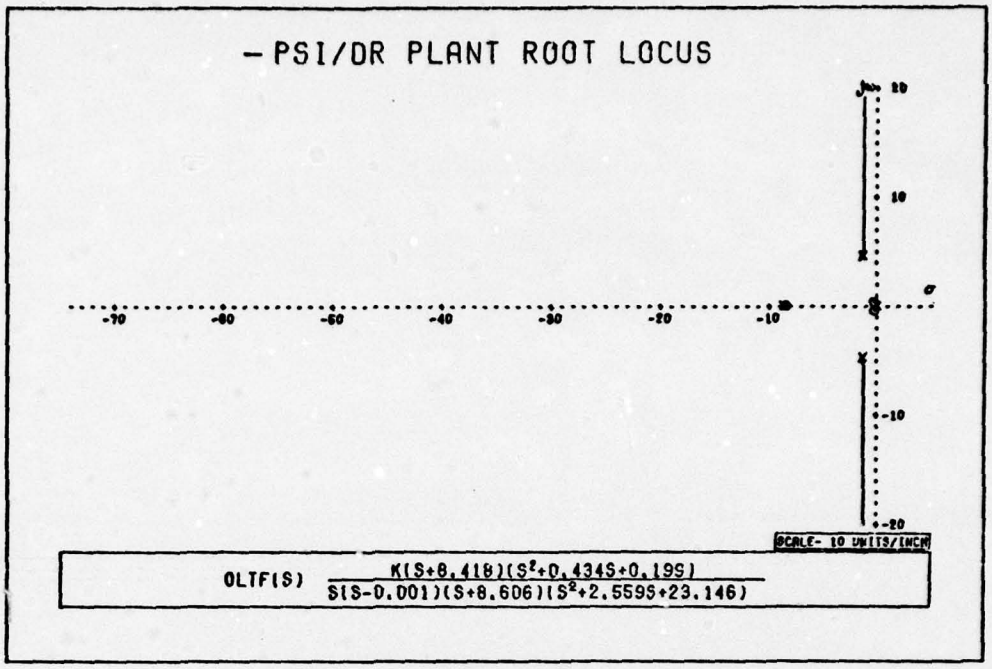


Figure A-38a: $-\psi/\delta_r$ Root Locus

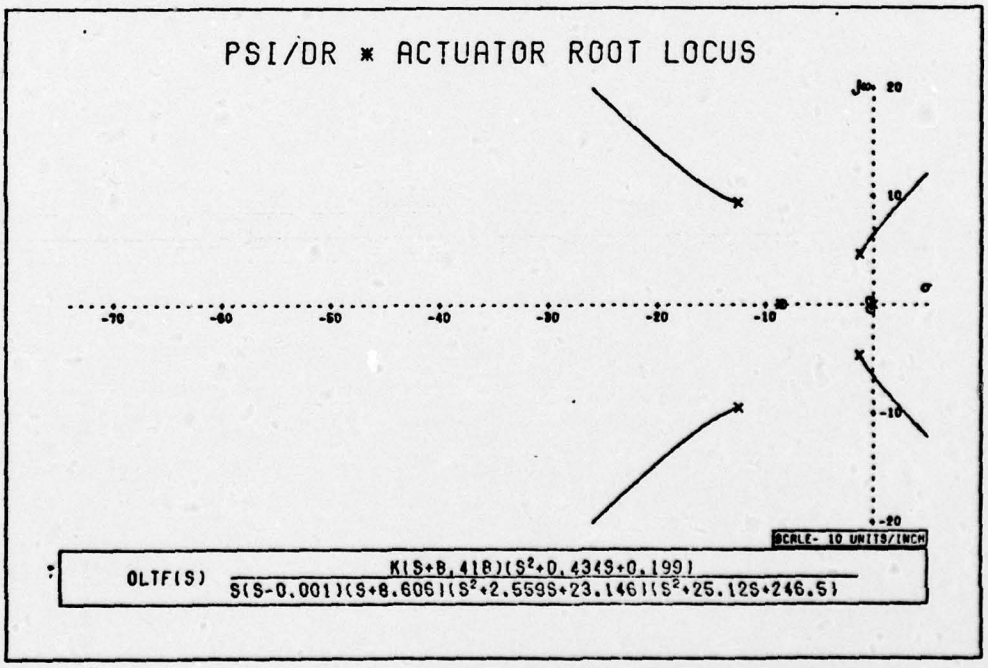


Figure A-38b: ψ/δ_r * Actuator Root Locus

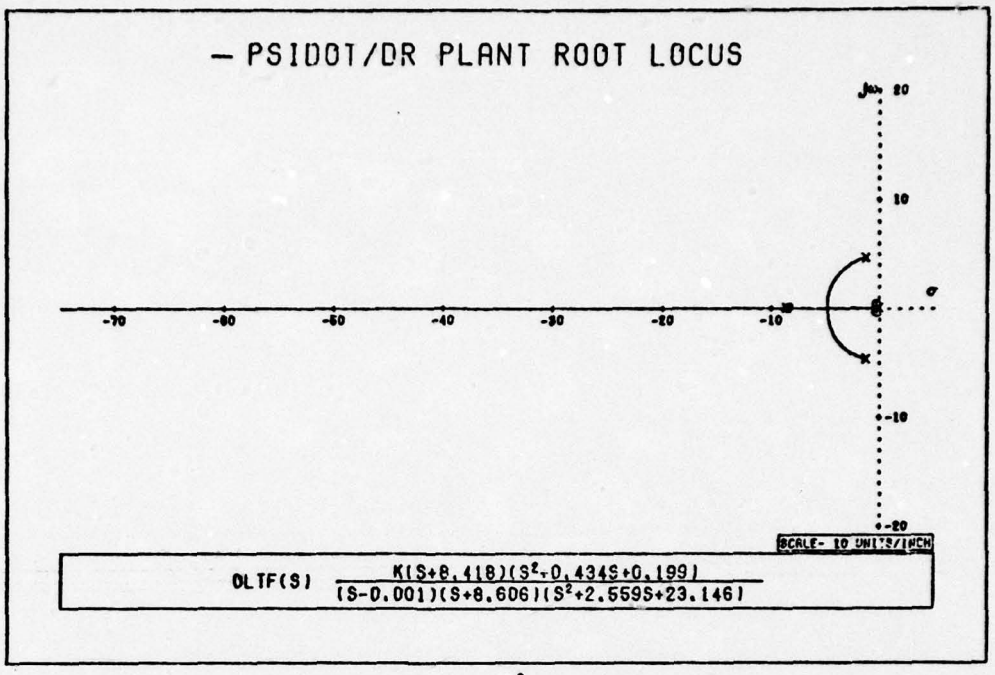


Figure A-39a: $-\dot{\psi}/\delta_r$ Root Locus

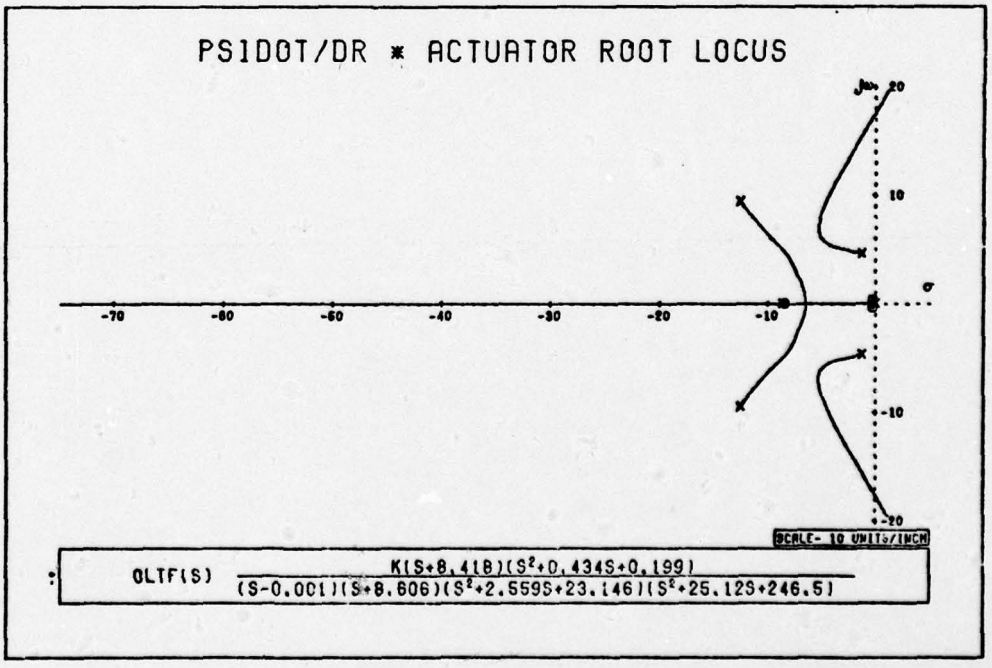


Figure A-39b: $\dot{\psi}/\delta_r$ * Actuator Root Locus

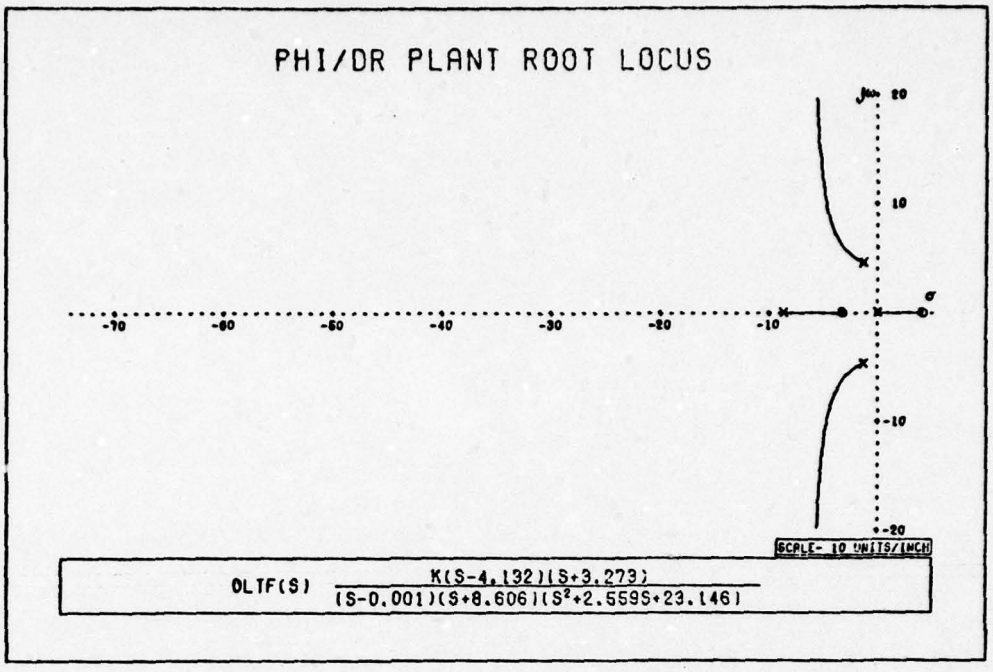


Figure A-40a: ϕ/δ_r Root Locus

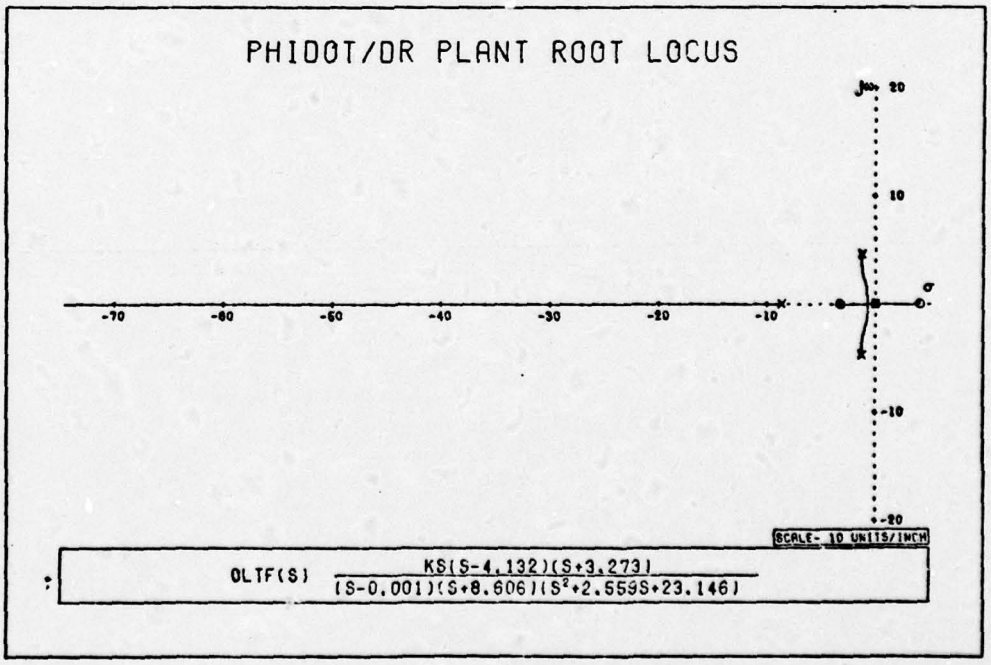


Figure A-40b: $\dot{\phi}/\delta_r$ Root Locus

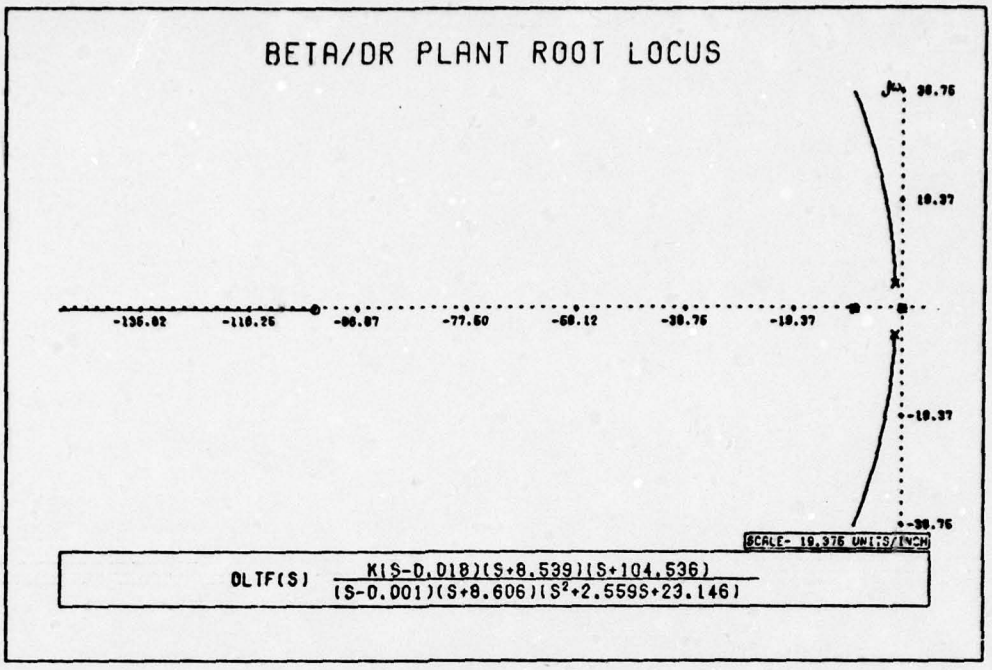


Figure A-41: β/δ_r Root Locus

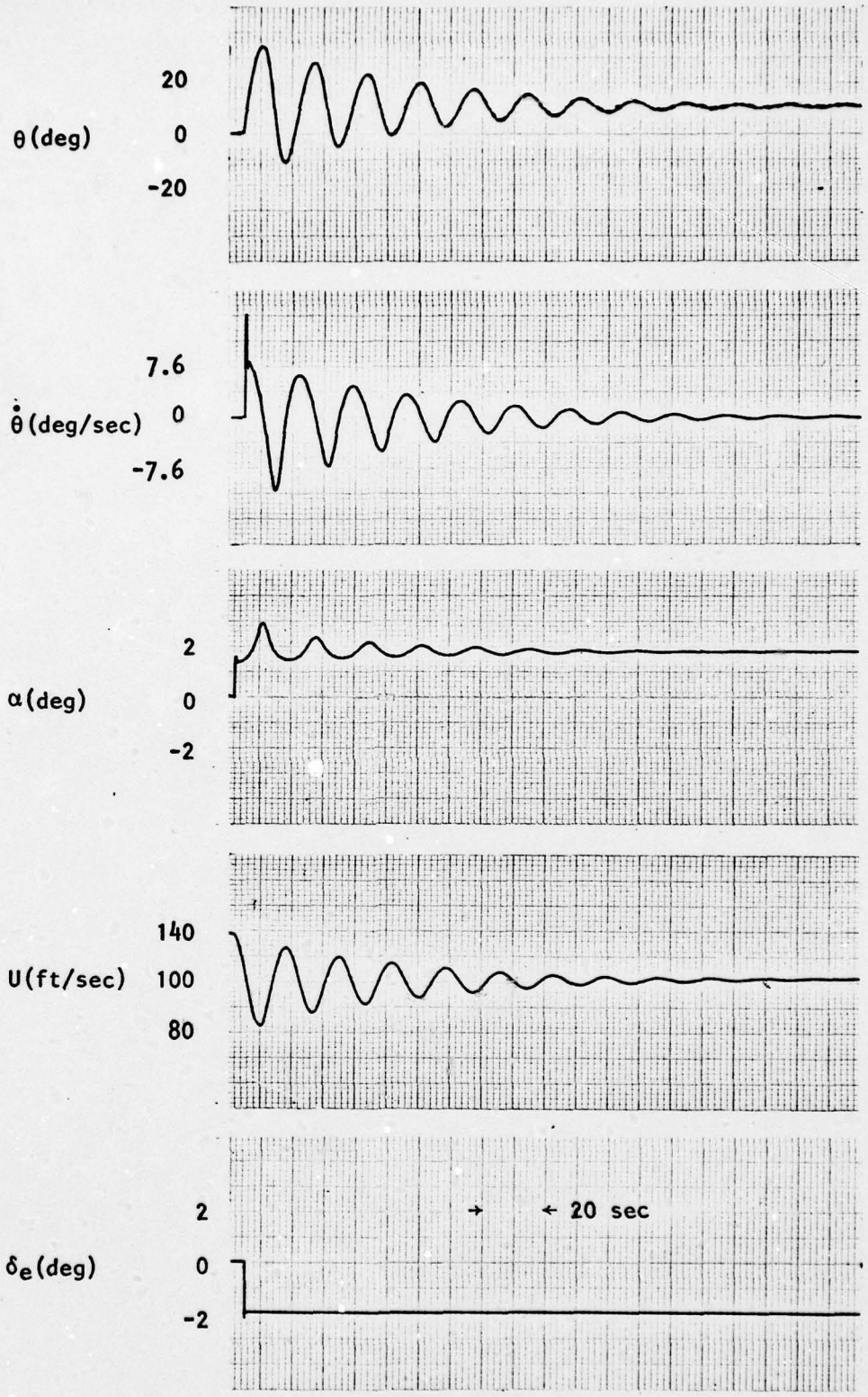


Figure A-42: Free RPV Airframe Response to a 2 Degree Step Elevator Deflection (Case II); Without Actuator

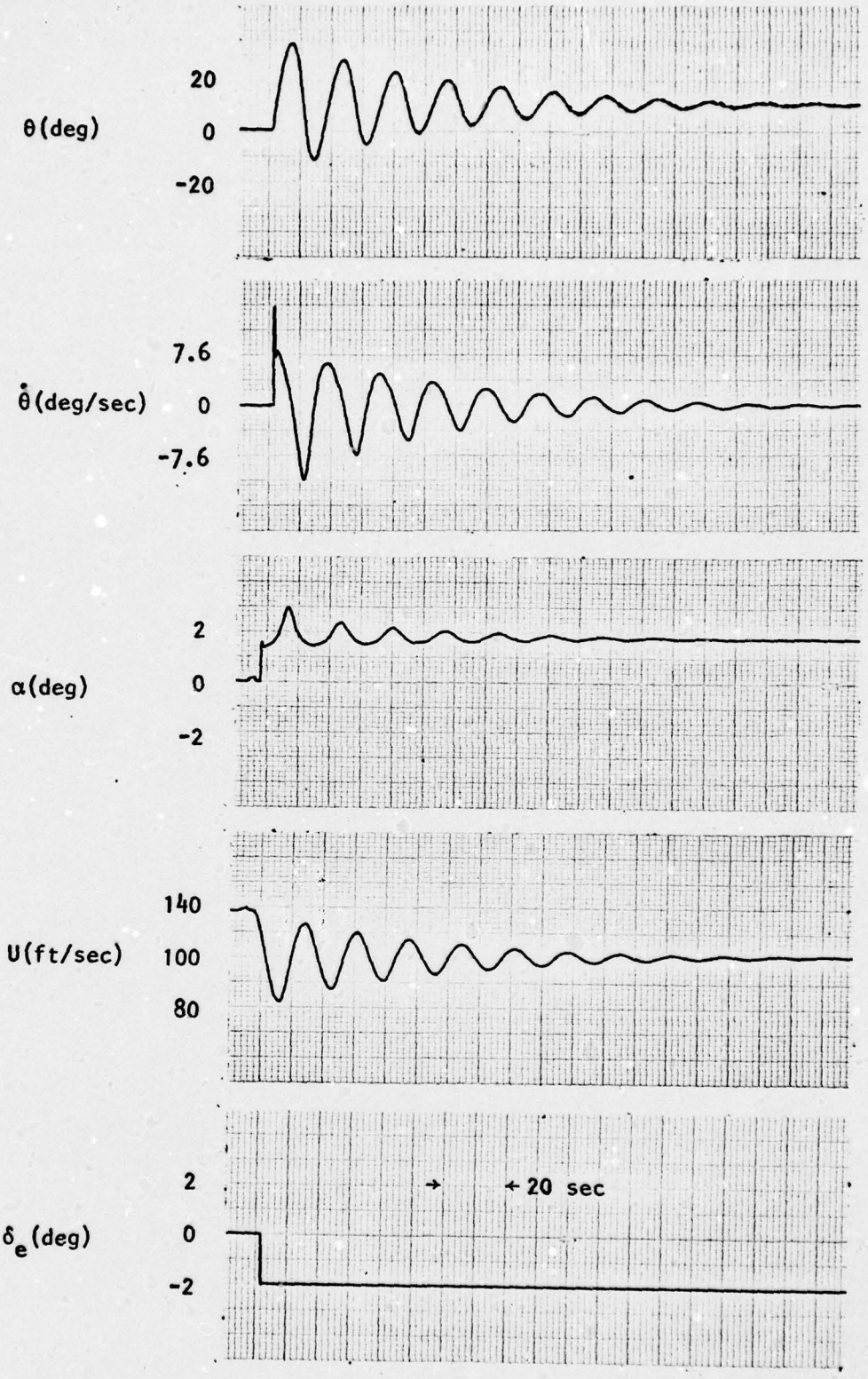


Figure A-43: Free RPV Airframe Response to a 2 Degree Step Elevator Deflection (Case II); With Actuator

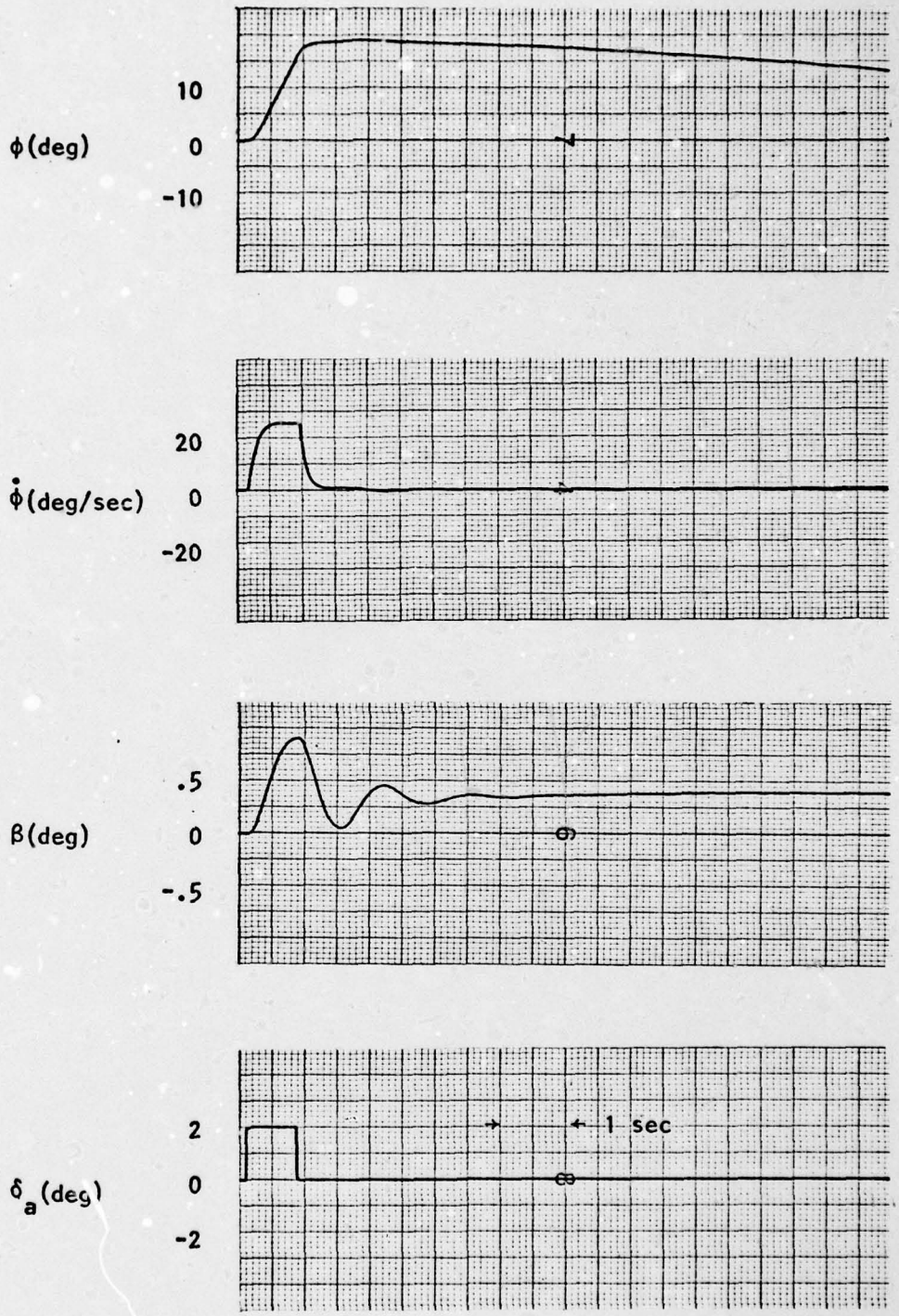


Figure A-44: Free RPV Airframe Response to a Pulse Aileron Deflection (Case II); Without Actuator

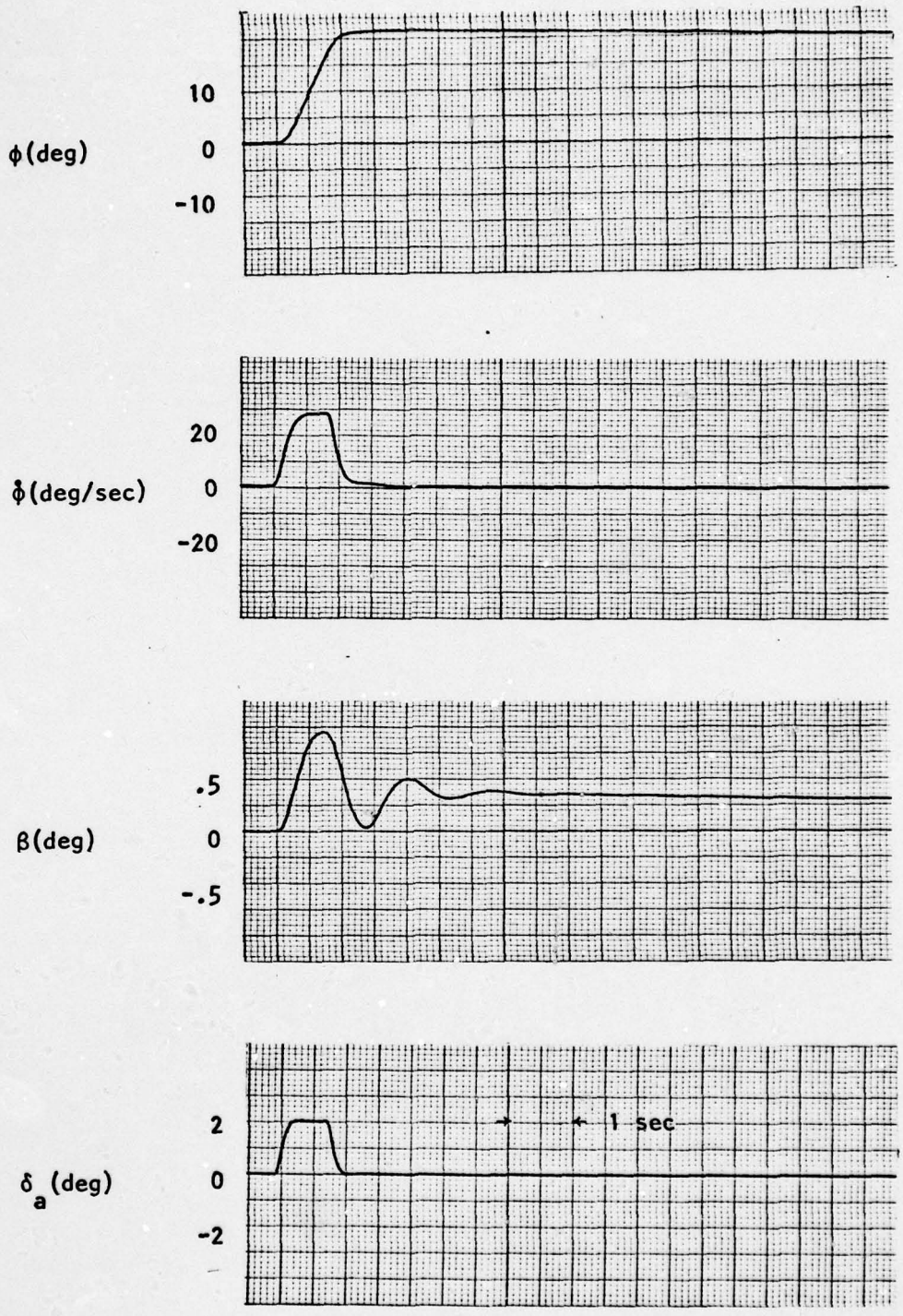


Figure A-45 : Free RPV Airframe Response to a Pulse Aileron Deflection (Case II); With Actuator

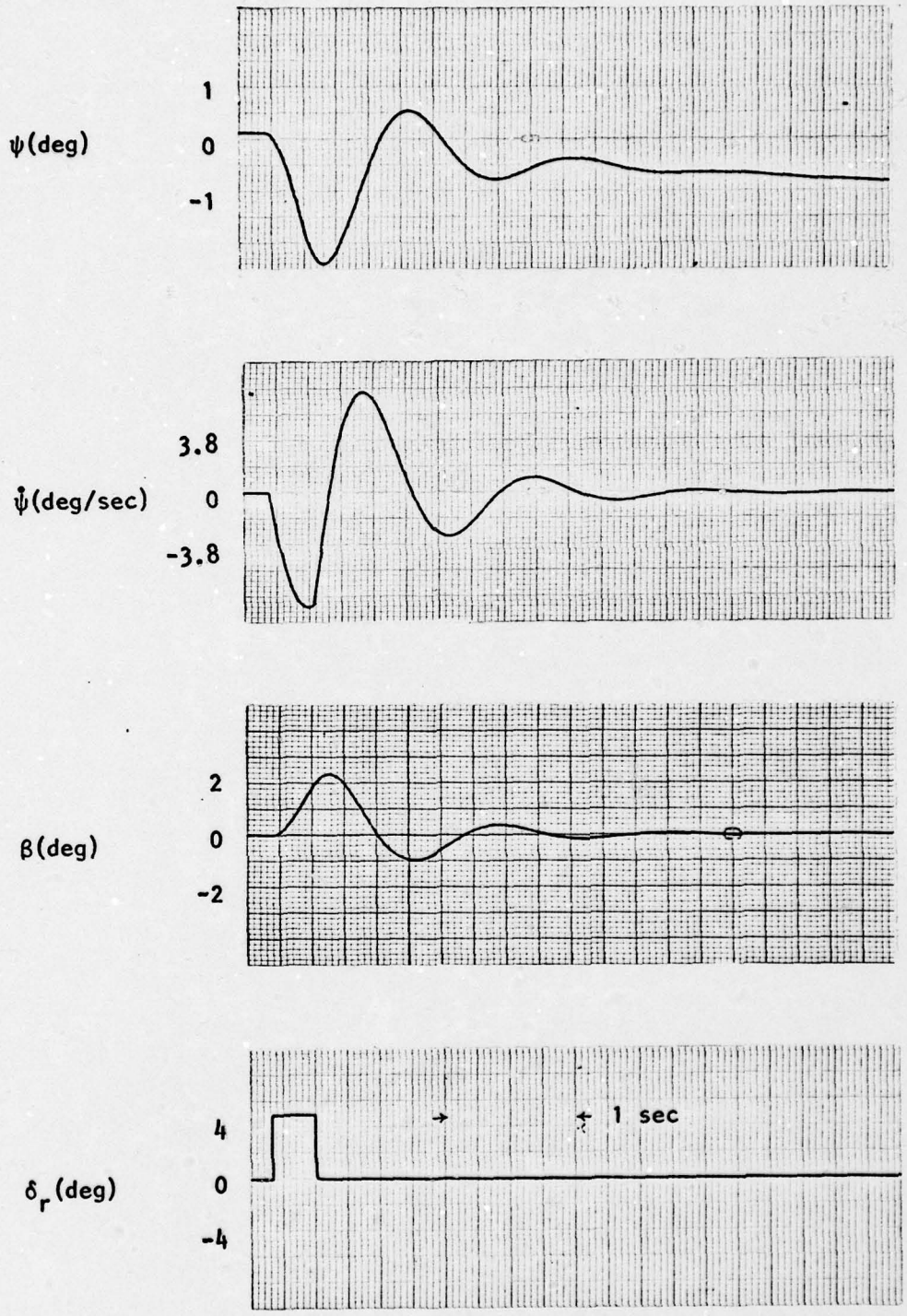


Figure A-46: Free RPV Airframe Response to a 5 Degree Pulse Rudder Deflection (Case II); Without Actuator

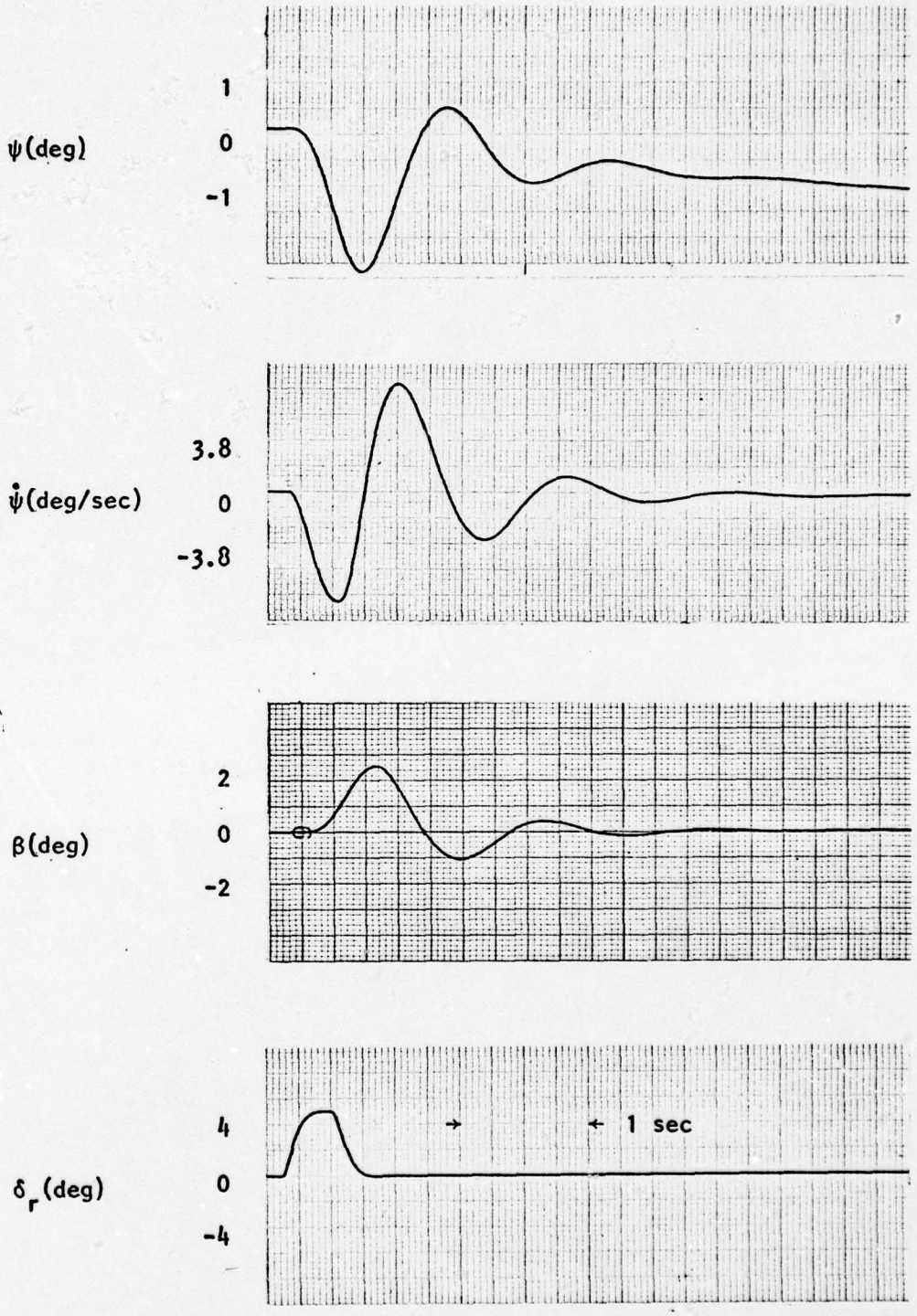


Figure A-47: Free RPV Airframe Response to a 5 Degree Pulse Rudder Deflection (Case II); With Actuator

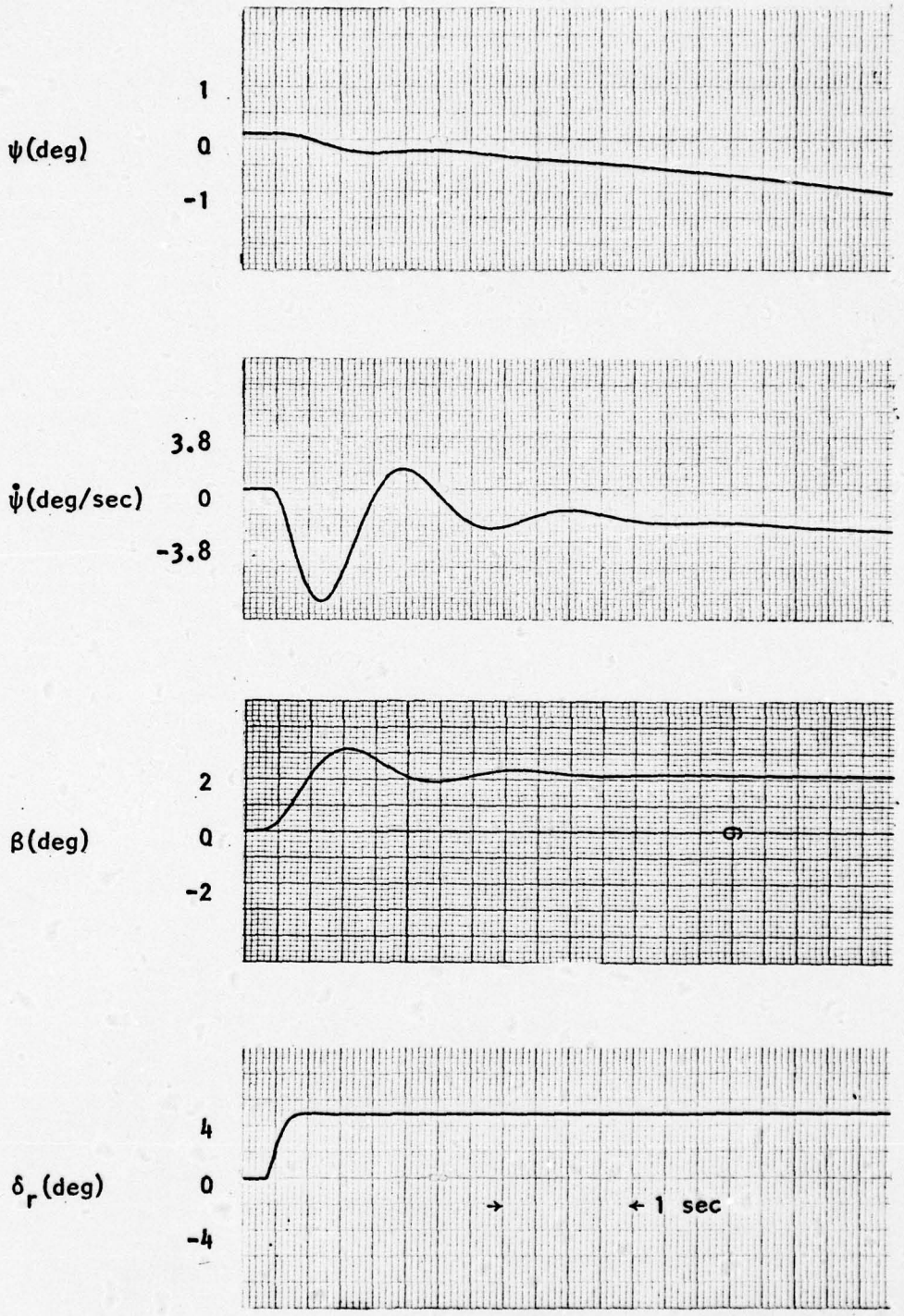


Figure A-48 : Free RPV Airframe Response to a 5 Degree Step Rudder Deflection (Case II); With Actuator

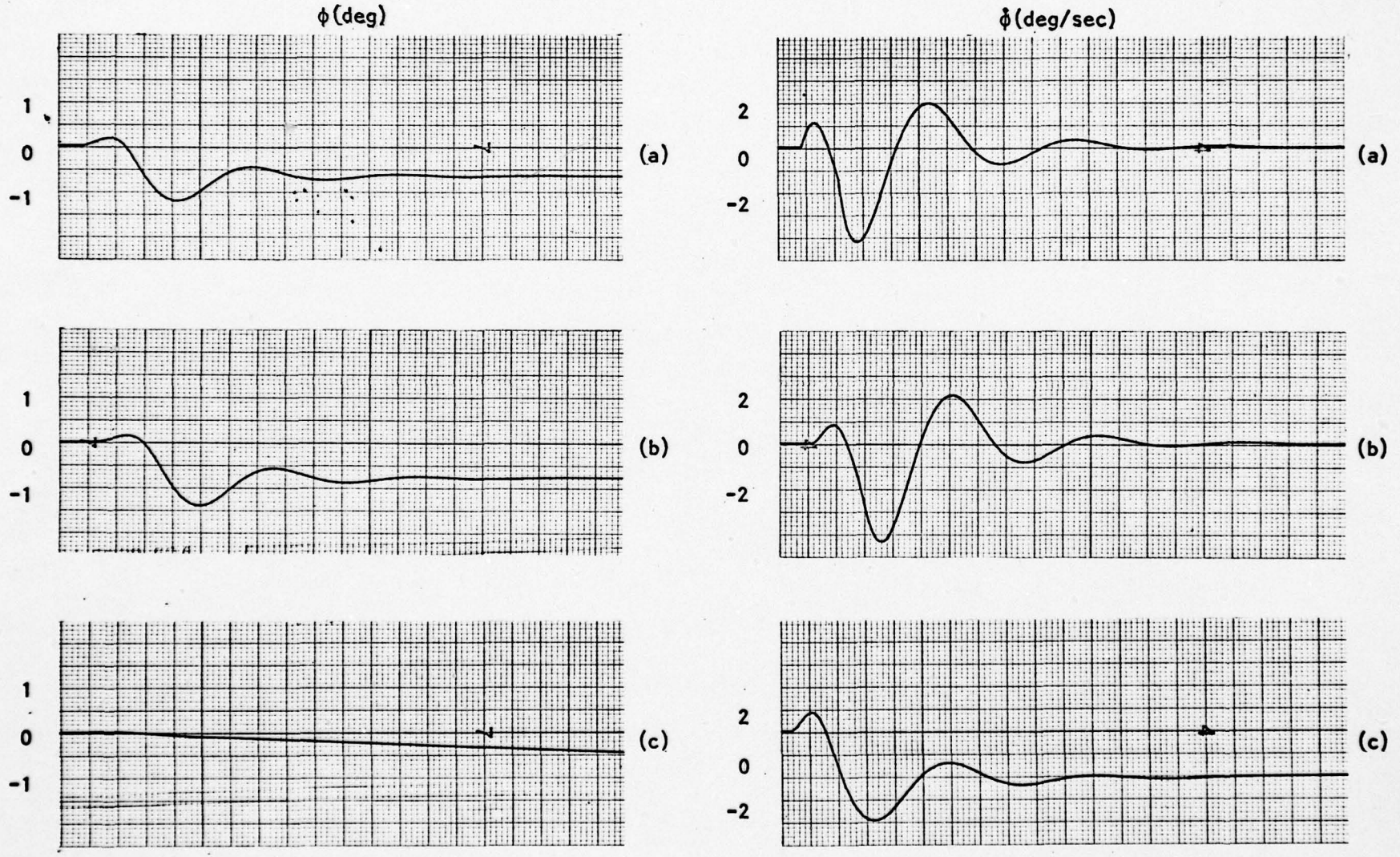


Figure A-49: Free RPV Airframe Response to a 5 Degree Rudder Deflection (Case II); (a) Without Actuator, Pulse (b) With Actuator, Pulse (c) With Actuator, Step

Appendix B

Autopilot Gain Descriptions

The purpose of this appendix is to provide the gains necessary to perform the A/P model analysis (chapter II) and digital A/P mechanization (chapter VI). Basic analog A/P feedback loop gains are given in (Ref 29) and were determined by manual tuning of a hybrid simulation analog A/P. Figure B-1 illustrates the general gain structure. This can be applied to any or all the various control loops. Note that all gains are mechanized in the feedback loops only. The electrical gains, those that can be tuned for performance specifications, are indicated by K_R and K_A, respectively. These were measured experimentally in the laboratory as a check of those listed in (Ref 29).

Gains for Linear Control Analysis (chapter II)

The control analysis of chapter II requires total DC loop gains. These are calculated by reference to Fig. B-1 as follows:

$$K_{RL} \text{ (Rate Loop)} = (K_{ACT})(K_R)(K_{RS})$$

$$K_{AL} \text{ (Attitude Loop)} = (K_{ACT})(K_A)(K_{VG})$$

Table B-1 provides a summary of these gains.

Gains for Digital A/P Controllers

The digital controllers require feedback gains for their implementation in software. These gains were measured electrically, directly from the analog A/P and follow the basic structure of Fig. B-2. Results are also included in table B-1.

TABLE B-1

Gains (Ref 29)

		Actuator Sensitivity (K_{ACT})	Vertical Gyro Sensitivity (K_{VG})	Rate Sensor Sensitivity (K_{RS})	Analog/Digital Feedback Gain (K_R, K_A)	TOTAL Loop Gains
Pitch	Rate	3.2 deg δ_e/V	-----	.05 V/deg θ/sec	.445 V/V	.0712 deg δ_e/deg θ/sec
	Attitude	3.2 deg δ_e/V	.04167 V/deg θ	-----	2.546 V/V	.3395 deg δ_e/deg θ
Roll	Rate	5 deg δ_a/V	-----	.05 V/degϕ/sec	.462 V/V	.116 deg δ_a/deg ϕ/sec
	Attitude	5 deg δ_a/V	.02778 V/deg	-----	1.922 V/V	.267 deg δ_a/deg ϕ
Yaw	Rate	6.4 deg δ_r/V	-----	.05 V/deg ψ/sec	1.83 V/V	.586 deg δ_r/deg ψ/sec

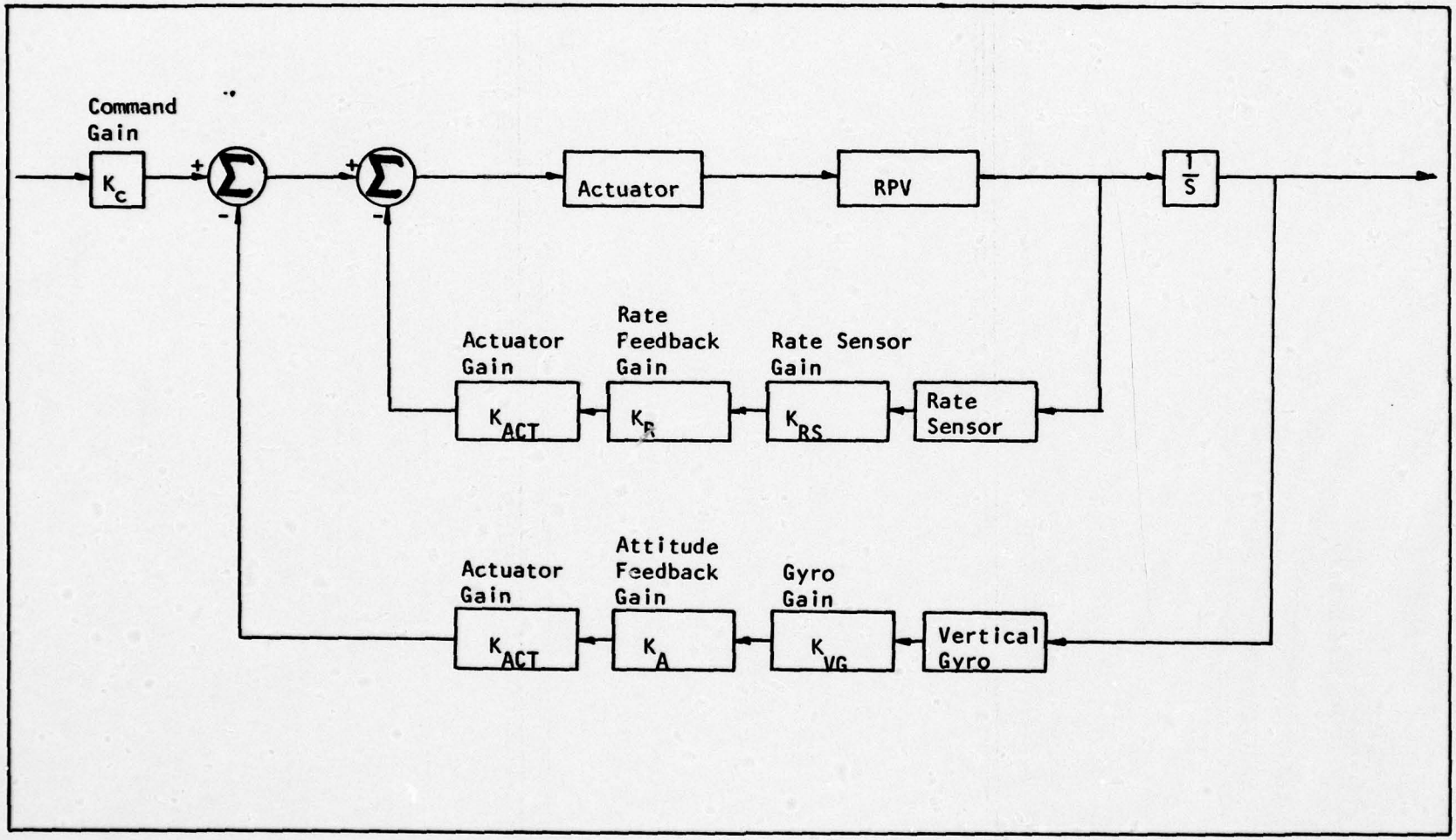


Figure B-1: Analog Autopilot Gain Structure (KBG Corp.) (Ref 29)

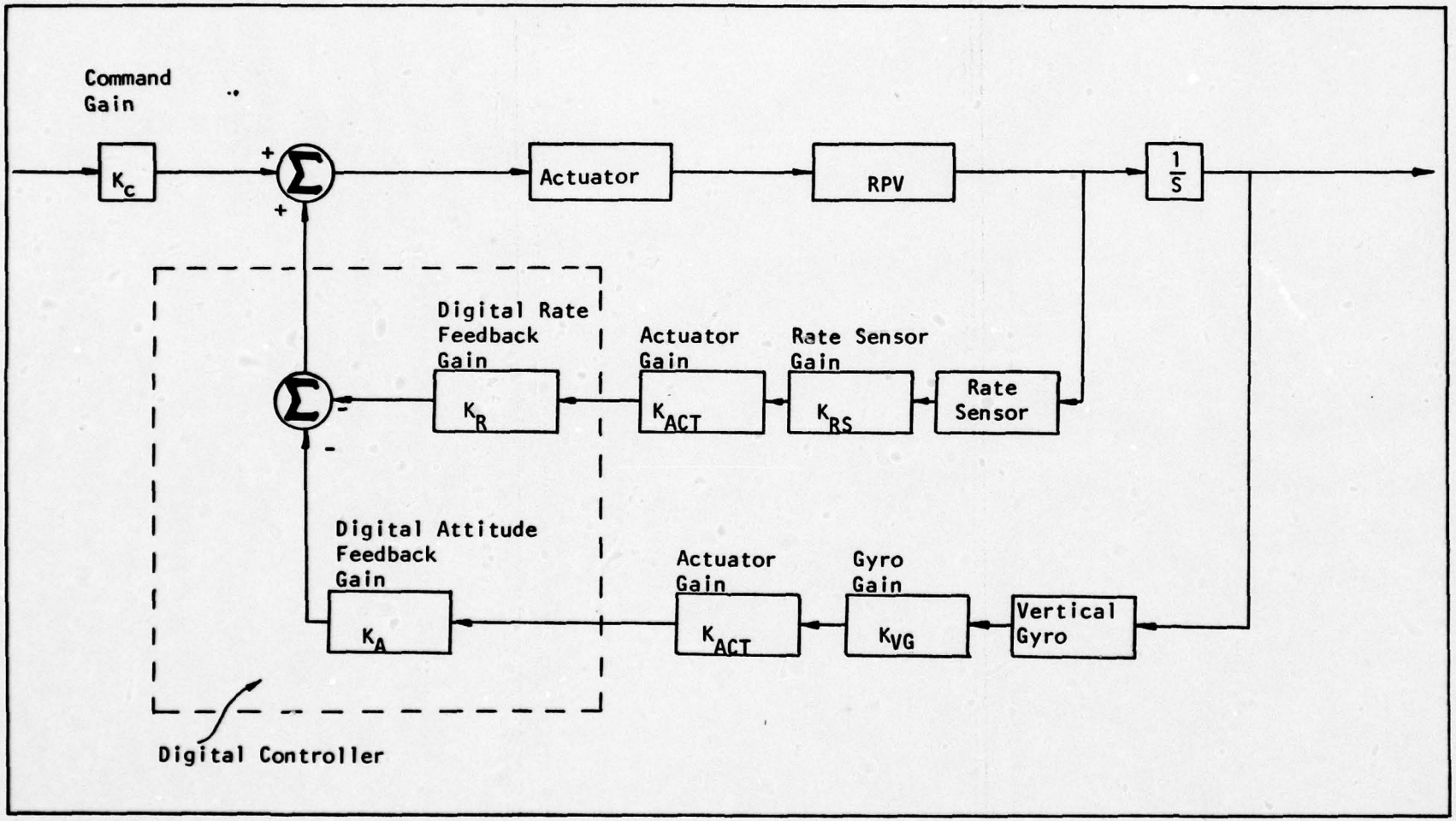
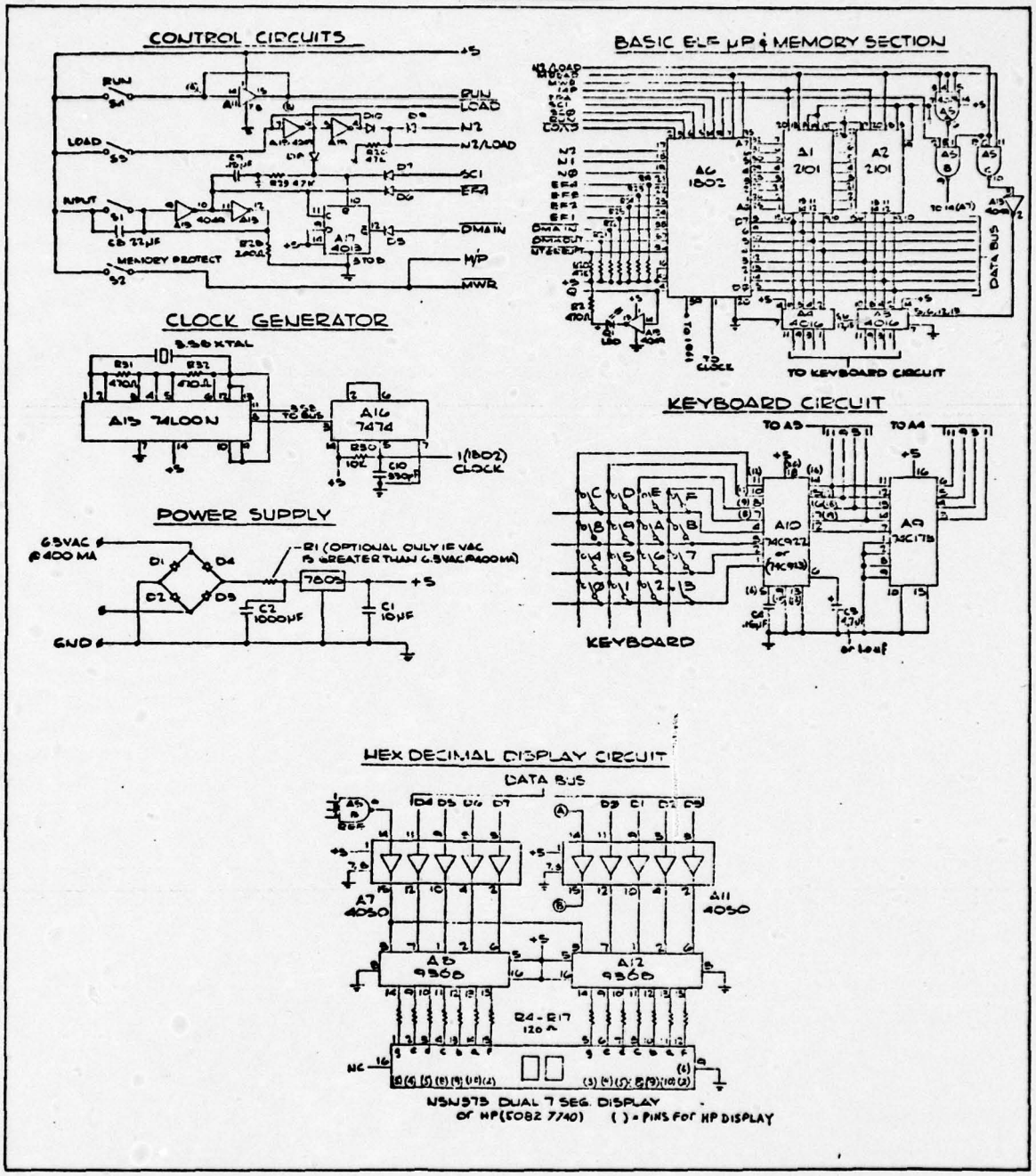


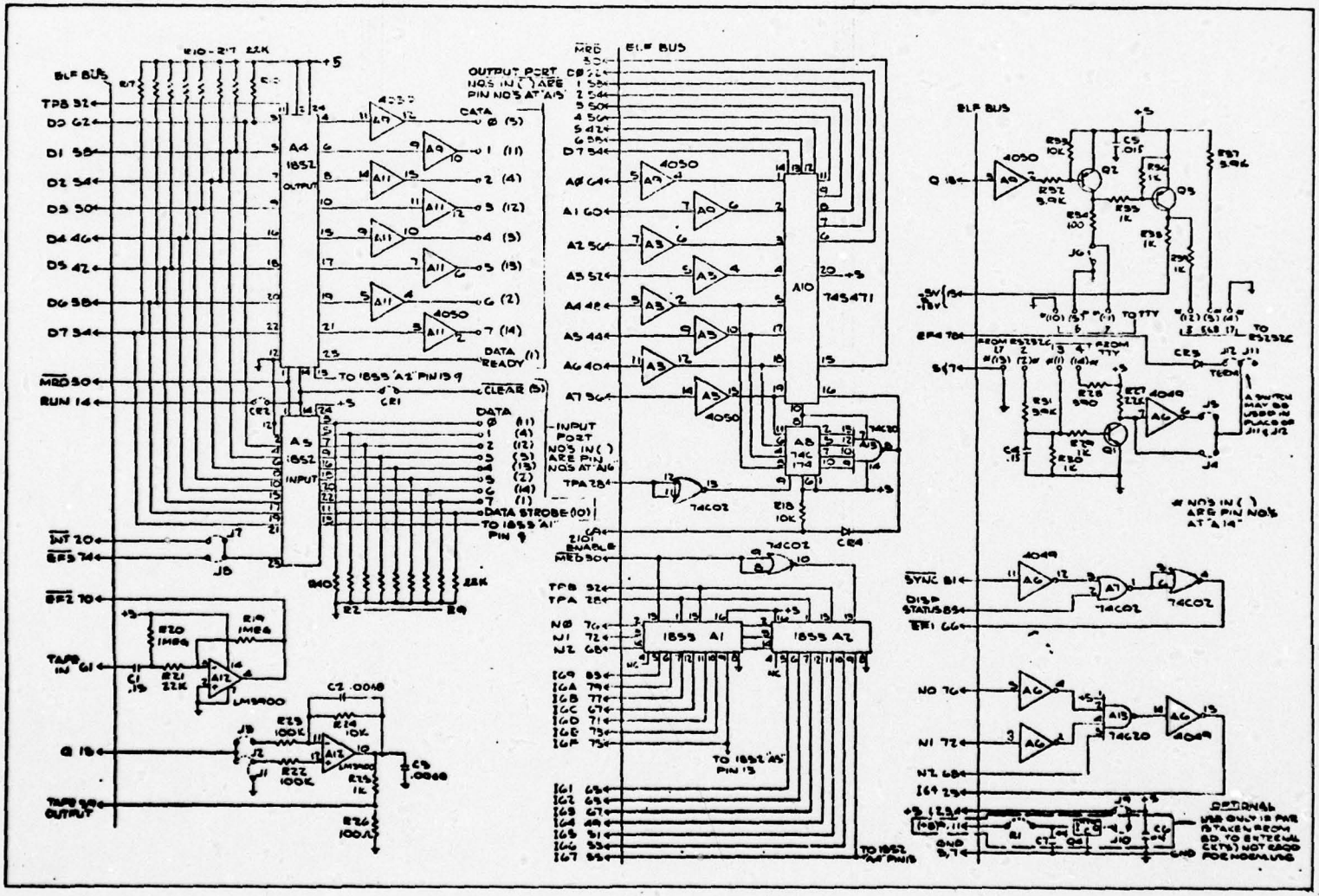
Figure B-2: Digital Autopilot Gain Structure

Appendix C

Schematics



Elf II Schematic



Giant Board Schematic

Appendix D

HP-29C Programmable Calculator Program to Examine Filter Characteristics on a Step-by-Step Basis

Line	Keycode	Step	Comments
01	15 13 01	gLBL 1	Initialize
02	01	1	Initialize
03	02	2	Initialize
04	08	8	Initialize
05	23 01	STO 1	Initialize
06	23 02	STO 2	Initialize
07	23 04	STO 4	Initialize
08	23 05	STO 5	Initialize
09	74	R/S	Stop and wait for input
10	15 13 02	gLBL 2	
11	23 00	STO 0	Store input, x (k)
12	24 06	RCL 6	Recall A_0
13	61	X	X (k) x A_0
14	24 01	RCL 1	Recall X (k-1)
15	24 07	RCL 7	Recall A_1
16	61	X	X (k-1) x A_1
17	51	+	
18	24 02	RCL 2	Recall X (k-2)
19	24 08	RCL 8	Recall A_2
20	61	X	X (k-2) x A_2
21	51	+	
22	24 04	RCL 4	Recall Y (k-1)
23	24 09	RCL 9	Recall B_1
24	61	X	Y (k-1) x B_1
25	51	+	
26	24 05	RCL 5	Recall Y (k-2)
27	24.0	RCL .0	Recall B_2
28	61	X	Y (k-2) x B_2
29	51	+	
30	06	6	÷64
31	04	4	÷64
32	71	÷	÷64
33	23 03	STO 3	Store Y (k)
34	24 04	RCL 4	Y (k-1)→Y (k-2)
35	23 05	STO 5	Y (k-1)→Y (k-2)
36	24 03	RCL 3	Y (k)→Y (k-1)
37	23 04	STO 4	Y (k)→Y (k-1)
38	24 01	RCL 1	X (k-1)→X (k-2)
39	23 02	STO 2	X (k-1)→X (k-2)
40	24 00	RCL 0	X (k)→X (k-1)
41	23 01	STO 1	X (k)→X (k-1)
42	24 03	RCL 3	Display Y (k)

43	74	R/S	Stop and wait for input
44	13 02	GTO 2	Go to label 2 (line 10)

Register Allocation:

R0 - X (k)	R1 - X (k-1)	R2 - X (k-2)
R3 - Y (k)	R4 - Y (k-1)	R5 - Y (k-2)
R6 - A_o	R7 - A_1	R8 - A_2
R9 - B_1	R.0 - B_2	

To Use:

----Load program.

----Store constants in R6 through R.0 (including sign).

----Depress 65B 1 to initialize.

----Enter input and depress R/S for each iteration. 128 represents 2.5 volts (center) and 256 represents 5 volts (full scale).

Appendix E

Digital Autopilot Program Listing

```
0000  ;          ..LISTING FOR XBQM-106 DIGITAL AUTOPILOT
0000  ;          ..FOR USE WITH CDP1802 MICROPROCESSOR
0000  ;          ..OCCUPIES 3/4 K BYTES
0000  C0F000     LBR 00 ..GO TO MONITOR
0003  F804       LDI 04 ..POINTER TO PAGE 4 FOR R1-R4, R6-RA
0005  B1         PHI 1 ..   "
0006  B2         PHI 2 ..   "
0007  B3         PHI 3 ..   "
0008  B4         PHI 4 ..   "
0009  B6         PHI 6 ..   "
000A  B7         PHI 7 ..   "
000B  B8         PHI 8 ..   "
000C  B9         PHI 9 ..   "
000D  BA         PHI A ..   "
000E  E6         SEX 6 .. SET RX = 6
000F  F8A9       LDI A9 .. A9→D
0011  A6         PLO 6 ..A9→R6.0 (ADDRESS OF TEMP)
0012  F801       LDI 01 .. 01→D
0014  56         STR 6 .. 01→M (R6)  (01→M(A9), 01 IS A/D MUX ADDRESS)
0015  67         OUT 7 .. LATCH MUX ADDRESS 01 AND START CONVERSION
0016  F8B9       LDI B9 .. B9→D
0018  A2         PLO 2 .. B9→R2.0 (LOCATION OF STACK)
0019  F8C0       LDI C0 .. C0→D
001B  A1         PLO 1 .. C0→R1.0 (LOCATION OF INPUT INTERRUPT ROUTINE)
001C  F8B0       LDI B0 .. B0→D
001E  AA         PLO A .. B0→RA.0 (LOCATION OF YAW RATE INPUT)
001F  A3         PLO 3 .. B0→R3 (LOCATION OF FIRST INPUT)
0020  F8CA       LDI CA .. CA→D
0022  A9         PLO 9 .. CA→R9.0 (LOCATION OF ADD SUBROUTINE)
0023  F8D6       LDI D6 .. D6→D
0025  A8         PLO 8 .. D6→R8.0 (LOCATION OF SUBTRACT SUBROUTINE)
0026  F8E2       LDI E2 .. E2→D
0028  A7         PLO 7 .. E2→R7.0 (LOCATION OF MULTIPLY SUBROUTINE)
0029  00         IDL .. WAIT FOR YAW RATE CONVERSION TO COMPLETE
002A  ;          ..
002A  ;          .. INITIALIZATION OF 2ND CONVERSION (YAW PENDULUM) FOLLOWS
002A  26         DEC 6 .. REPOSITION POINTER TO TEMP (A9)
002B  F802       LDI 02 .. 02→D
002D  56         STR 6 .. 02→M (R6) (02 →M(A9), 02 IS A/D MUX ADDRESS)
002E  67         OUT 7 .. LATCH MUX ADDRESS 02 AND START CONVERSION
002F  F8AB       LDI AB .. AB → D
0031  AA         PLO A .. AB → RA.0 (LOCATION OF YAW PENDULUM INPUT)
0032  ;          ..
0032  ;:         .. A0 • X(k) FOLLOWS
0032  F8BB       LDI BB .. BB → D
0034  A4         PLO 4 .. BB → R4.0 (POINTER TO OPERAND 1)
0035  03         LDN 3 .. RETRIEVE X(k) (YAW RATE)
0036  54         STR 4 .. X(k) → M(R4) (X(k) → OPERAND 1)
```

```
0037  F8BA      LDI BA .. BA → D
0039  A3        PLO 3 .. BA → R3.0 (POINTER TO OPERAND 2)
003A  F83F      LDI 3F .. 3F → D
003C  53        STR 3 .. 3F → M(R3) (3F → OPERAND 2)
003D  D7        SEP 7 .. GO TO MULTIPLY SUBROUTINE
003E  ;         ..
003E  ;         .. INITIALIZATION OF 3RD CONVERSION (ROLL RATE) FOLLOWS
003E  E6        SEX 6 .. RESET X
003F  26        DEC 6 .. REPOSITION POINTER TO TEMP (A9)
0040  F803      LDI 03 .. 03 → D
0042  56        STR 6 .. 03 → M(R6)(03 →M(A9), 03 IS A/D MUX ADDRESS)
0043  67        OUT 7 .. LATCH MUX ADDRESS 03 AND START CONVERSION
0044  1A        INC A .. INCREMENT INPUT LOCATION
0045  ;         ..
0045  ;         .. A2 · X(k-2) FOLLOWS
0045  F8B2      LDI B2 .. B2 → D
0047  A3        PLO 3 .. B2 → R3.0 (POINTER TO X(k-2))
0048  F8BD      LDI BD .. BD → D
004A  A4        PLO 4 .. BD → R4.0 (POINTER TO OPERAND 1)
004B  03        LDN 3 .. RETRIEVE X(k-2)
004C  54        STR 4 .. X(k-2) → M(R4), (X(k-2) → OPERAND 1)
004D  F8BC      LDI BC .. BC→ D
004F  A3        PLO 3 .. BC → R3.0 (POINTER TO OPERAND 2)
0050  F83F      LDI 3F .. 3F → D
0052  53        STR 3 .. 3F → M(R3) (3F → OPERAND 2)
0053  D7        SEP 7 .. GO TO MULTIPLY SUBROUTINE
0054  ;         ..
0054  ;         .. SUMMATION OF A0 · X(k) + A2 · X(k-2) FOLLOWS
0054  23        DEC 3 .. REPOSITION OPERAND POINTER
0055  D9        SEP 9 .. GO TO ADD SUBROUTINE
0056  ;         ..
0056  ;         .. INITIALIZATION OF 4TH CONVERSION (ROLL GYRO) FOLLOWS
0056  E6        SEX 6 .. RESET X
0057  26        DEC 6 .. REPOSITION POINTER TO TEMP (A9)
0058  F804      LDI 04 .. 04 → D
005A  56        STR 6 .. 04 → M(R6), (04 → M(A9), 04 IS A/D MUX ADDRESS)
005B  67        OUT 7 .. LATCH MUX ADDRESS 04 AND START CONVERSION
005C  1A        INC A .. INCREMENT INPUT LOCATION
005D  ;         ..
005D  ;         .. B1 · Y(k-1) FOLLOWS
005D  F8B3      LDI B3 .. B3 → D
005F  A4        PLO 4 .. B3 → R4.0 (POINTER TO Y(k-1))
0060  04        LDN 4 .. RETRIEVE Y(k-1)
0061  53        STR 3 .. Y(k-1) → M(R3), (Y(k-1) → OPERAND 1)
0062  F8BB      LDI BB .. BB → D
0064  A4        PLO 4 .. BB → R4.0 (POINTER TO OPERAND 2)
0065  F87E      LDI 7E .. 7E → D
0067  54        STR 4 .. 7E → M(R4) (7E → OPERAND 2)
0068  D7        SEP 7 .. GO TO MULTIPLY SUBROUTINE
0069  ;         ..
0069  ;         .. PREVIOUS SUMMATION + B1 · Y(k-1) FOLLOWS
0069  13        INC 3 .. POSITION POINTER TO OPERAND 1
```

```
006A  14        INC 4 .. POSITION POINTER TO OPERAND 2
006B  14        INC 4 ..  "
006C  D9        SEP 9 .. GO TO ADD SUBROUTINE
006D  ;         ..
006D  ;         .. INITIALIZATION OF 5TH CONVERSION (PITCH RATE) FOLLOWS
006D  E6        SEX 6 .. RESET X
006E  26        DEC 6 .. REPOSITION POINTER TO TEMP (A9)
006F  F805      LDI 05 .. 05 → D
0071  56        STR 6 .. 05 → M(R6), (05 → M(A9), 05 IS A/D MUX ADDRESS)
0072  67        OUT 7 .. LATCH MUX ADDRESS 05 AND START CONVERSION
0073  1A        INC A .. INCREMENT INPUT LOCATION
0074  ;         ..
0074  ;         .. B1 · Y(k-1) FRAC FOLLOWS
0074  F8B5      LDI B5 .. B5 → D
0076  A4        PLO 4 .. B5 → R4.0 (POINTER TO Y(k-1) FRAC)
0077  04        LDN 4 .. RETRIEVE Y(k-1) FRAC
0078  53        STR 3 .. Y(k-1) FRAC→ M(R3), (Y(k-1) FRAC → OPERAND 1)
0079  F8BB      LDI BB .. BB → D
007B  A4        PLO 4 .. BB → R4.0 (POINTER TO OPERAND 2)
007C  F87E      LDI 7E .. 7E → D
007E  54        STR 4 .. 7E → M(R4) (7E → OPERAND 2)
007F  D7        SEP 7 .. GO TO MULTIPLY SUBROUTINE
0080  ;         ..
0080  ;         .. DIVISION OF ABOVE PRODUCT BY 8 FOLLOWS (SHIFT 8 PLACES
0080  ;         .. RIGHT)
0080  03        LDN 3 .. RETRIEVE 8 MSB'S OF PRODUCT
0081  54        STR 4 .. PLACE INTO 8 LSB'S OF PRODUCT
0082  F800      LDI 00 .. 00 → D
0084  53        STR 3 .. ZERO INTO 8 MSB'S OF PRODUCT
0085  ;         ..
0085  ;         .. PREVIOUS SUMMATION + B1 . Y(k-1) FRAC FOLLOWS
0085  13        INC 3 .. POSITION POINTER TO OPERAND 1
0086  14        INC 4 .. POSITION POINTER TO OPERAND 2
0087  14        INC 4 ..  "
0088  D9        SEP 9 .. GO TO ADD SUBROUTINE
0089  ;         ..
0089  ;         ..INITIALIZATION OF 6TH CONVERSION (PITCH GYRO) FOLLOWS
0089  E6        SEX 6 .. RESET X
008A  26        DEC 6 .. REPOSITION POINTER TO TEMP (A9)
008B  F806      LDI 06 .. 06 →  D
008D  56        STR 6 .. 06 → M(R6), (06 → M(A9), 06 IS A/D MUX ADDRESS)
008E  67        OUT 7 .. LATCH MUX ADDRESS 06 AND START CONVERSION
008F  1A        INC A .. INCREMENT INPUT LOCATION
0090  ;         ..
0090  ;         .. A1 · X(k-1) FOLLOWS
0090  F8B1      LDI B1 .. B1 → D
0092  A4        PLO 4 .. B1 → R4.0 (POINTER TO X(k-1))
0093  04        LDN 4 .. RETRIEVE X(k-1)
0094  53        STR 3 .. X(k-1) → M(R3), (X(k-1) → OPERAND 1)
0095  F8BB      LDI BB .. BB → D
0097  A4        PLO 4 .. BB → R4.0, (POINTER TO OPERAND 2)
```

```
0098  F87E      LDI 7E .. 7E → D
009A  54        STR 4 .. 7E → M(R4), (7E → OPERAND 2)
009B  D7        SEP 7 .. GO TO MULTIPLY SUBROUTINE
009C  ;         ..
009C  ;         .. PREVIOUS SUMMATION - A1 · X(k-1) FOLLOWS
009C  13        INC 3 .. POSITION POINTER TO OPERAND 1
009D  14        INC 4 .. POSITION POINTER TO OPERAND 2
009E  14        INC 4 .. "
009F  D8        SEP 8 .. GO TO SUBTRACT SUBROUTINE
00A0  ;         ..
00A0  3BE8      BNF E8 .. GO TO OVERFLOW IF SUBTRACTION RESULT NEG
00A2  ;         ..
00A2  ;         .. B2 · Y(k-2) FOLLOWS
00A2  F8B4      LDI B4 .. B4 → D
00A4  A4        PLO 4 .. B4 → R4.0, (POINTER TO Y(k-2))
00A5  04        LDN 4 .. RETRIEVE Y(k-2)
00A6  53        STR 3 .. Y(k-2) → M(R3), (Y(k-2) → OPERAND 1)
00A7  F8BB      LDI BB .. BB → D
00A9  A4        PLO 4 .. BB → R4.0 (POINTER TO OPERAND 2)
00AA  F83E      LDI 3E .. 3E → D
00AC  54        STR 4 .. 3E → M(R4), (3E → OPERAND 2)
00AD  D7        SEP 7 .. GO TO MULTIPLY SUBROUTINE
00AE  ;         ..
00AE  ;         .. PREVIOUS SUMMATION - B2 · Y(k-2) FOLLOWS
00AE  13        INC 3 .. POSITION POINTER TO OPERAND 1
00AF  14        INC 4 .. POSITION POINTER TO OPERAND 2
00B0  14        INC 4 .. "
00B1  D8        SEP 8 .. GO TO SUBTRACT SUBROUTINE
00B2  ;         ..
00B2  3BE8      BNF E8 .. GO TO OVERFLOW IF SUBTRACTION RESULT NEG
00B4  ;         ..
00B4  ;         .. B2 · Y(k-2) FRAC FOLLOWS
00B4  F8B6      LDI B6 .. B6 → D
00B6  A4        PLO 4 .. B6 → R4.0, (POINTER TO Y(k-2)FRAC)
00B7  04        LDN 4 .. RETRIEVE Y(k-2) FRAC
00B8  53        STR 3 .. Y(k-2) FRAC → M(R3), (Y(k-2) FRAC → OPERAND 1)
00B9  F8BB      LDI BB .. BB → D
00BB  A4        PLO 4 .. BB → R4.0, (POINTER TO OPERAND 2)
00BC  F83B      LDI 3E .. 3E → D
00BE  54        STR 4 .. 3E → M(R4), (3E → OPERAND 2)
00BF  D7        SEP 7 .. GO TO MULTIPLY SUBROUTINE
00C0  ;         ..
00C0  ;         .. DIVISION OF ABOVE PRODUCT BY 8 FOLLOWS (SHIFT 8 PLACES
00C0  ;         .. RIGHT)
00C0  03        LDN 3 .. RETRIEVE 8 MSB'S OF PRODUCT
00C1  54        STR 4 .. PLACE INTO 8 LSB'S OF PRODUCT
00C2  F800      LDI 00 .. 00 → D
00C4  53        STR 3 .. ZERO INTO 8 MSB'S OF PRODUCT
00C5  ;         ..
00C5  ;         .. PREVIOUS SUMMATION - B2 · Y(k-2) FRAC FOLLOWS
00C5  13        INC 3 .. POSITION POINTER TO OPERAND 1
```

```
00C6  14        INC 4 .. POSITION POINTER TO OPERAND 2
00C7  14        INC 4 ..  "
00C8  D8        SEP 8 .. GO TO SUBTRACT SUBROUTINE
00C9  ;         ..
00C9  3BE8      BNF E8 .. GO TO OVERFLOW IF SUBTRACTION RESULT NEG
00CB  ;         ..
00CB  ;         .. CHECK FOR EXCESSIVE SUM (A ONE IN 2 MSB'S) FOLLOWS
00CB  04        LDN 4 .. RETRIEVE SUM
00CC  FE        SHL .. SHIFT LEFT (MSB →DF)
00CD  33E8      BDF E8 .. GO TO OVERFLOW IF DF=1
00CF  FE        SHL .. SHIFT LEFT (MSB-1 → DF)
00D0  33E8      BDF E8 .. GO TO OVERFLOW IF DF=1
00D2  ;         ..
00D2  ;         .. REPOSITION OF VALUES OF NEXT ITERATION FOLLOWS
00D2  F8B5      LDI B5 .. B5 →  D
00D4  A3        PLO 3 .. B5 → R3.0 (POINTER TO Y(k-1) FRAC)
00D5  43        LDA 3 .. RETRIEVE Y(k-1) FRACT
00D6  53        STR 3 .. Y(k-1) FRAC→  Y(k-2) FRAC
00D7  23        DEC 3 .. REPOSITION POINTER TO Y(k-1)
00D8  23        DEC 3 ..  "
00D9  23        DEC 3 ..  "
00DA  43        LDA 3 .. RETRIEVE Y(k-1)
00DB  53        STR 3 .. Y(k-1) → Y(k-2)
00DC  23        DEC 3 .. REPOSITION POINTER TO X(k-1)
00DD  23        DEC 3 ..  "
00DE  23        DEC 3 ..  "
00DF  43        LDA 3 .. RETRIEVE X(k-1)
00E0  53        STR 3 .. X(k-1)  →X(k-2)
00E1  23        DEC 3 .. REPOSITION POINTER TO X(k)
00E2  23        DEC 3 ..  "
00E3  43        LDA 3 .. RETRIEVE X(k)
00E4  53        STR 3 .. X(k) →  X(k-1)
00E5  C00100    LBR 0100 .. GO TO NEXT MEMORY PAGE (PAGE 01)
00E8  ;         ..
00E8  ;         .. OVERFLOW ROUTINE FOLLOWS
00E8  F8B6      LDI B6 .. B6 →  D
00EA  A3        PLO 3 .. B6 → R3.0, (POINTER TO Y(k-2) FRAC)
00EB  F800      LDI 00 .. 00 →  D
00ED  53        STR 3 .. 00 → M(R3), (00 →  Y(k-2) FRAC)
00EE  23        DEC 3 .. REPOSITION POINTER TO Y(k-1) FRAC
00EF  53        STR 3 .. 00 → Y(k-1) FRAC
00F0  23        DEC 3 .. REPOSITION POINTER TO Y(k-2)
00F1  F880      LDI 80 .. 80 →  D
00F3  53        STR 3 .. 80 → M(R3), (80 → Y(k-2))
00F4  23        DEC 3 .. REPOSITION POINTER TO Y(k-1)
00F5  53        STR 3 .. 80 → Y(k-1)
00F6  23        DEC 3 .. REPOSITION POINTER TO X(k-2)
00F7  53        STR 3 .. 80 → X(k-2)
00F8  23        DEC 3 .. REPOSITION POINTER TO X(k-1)
00F9  53        STR 3 .. 80 → X(k-1)
00FA  23        DEC 3 .. REPOSITION POINTER TO X(k)
00FB  E6        SEX 6 .. RESET X
```

```
00FC  302A      BR 2A .. GO TO 002A (THRU FILTER ALGORITHM AGAIN)
00FE  00        IDL .. USED BY MONITOR
00FF  00        IDL .. USED BY MONITOR
0100  ;         ..
0100  ;         .. DIVISION BY 64 (BY BINARY POINT SHIFTING) FOLLOWS
0100  F8BA      LDI BA .. BA → D
0102  AA        PLO A .. BA → RA.0, (POINTER TO UPPER BYTE OF RESULT)
0103  F8B5      LDI B5 .. B5 → D
0105  A3        PLO 3 .. B5 → R3.0 (POINTER TO Y(k-1) FRAC)
0106  14        INC 4 .. POSITION POINTER TO LOWER BYTE OF RESULT
0107  04        LDN 4 .. RETRIEVE LOWER BYTE
0108  FE        SHL .. SHIFT LEFT
0109  54        STR 4 .. REPLACE LOWER BYTE
010A  24        DEC 4 .. POSITION POINTER TO UPPER BYTE
010B  04        LDN 4 .. RETRIEVE UPPER BYTE
010C  7E        SHLC .. SHIFT LEFT WITH CARRY-IN
010D  54        STR 4 .. REPLACE UPPER BYTE
010E  14        INC 4 .. POSITION POINTER TO LOWER BYTE
010F  04        LDN 4 .. RETRIEVE LOWER BYTE
0110  FE        SHLC .. SHIFT LEFT
0111  53        STR 3 .. LOWER BYTE → Y(k-1) FRAC
0112  23        DEC 3 .. POSITION POINTER TO Y(k-1)
0113  23        DEC 3 ..  "
0114  24        DEC 4 .. POSITION POINTER TO UPPER BYTE
0115  04        LDN 4 .. RETRIEVE UPPER BYTE
0116  7E        SHLC .. SHIFT LEFT WITH CARRY-IN
0117  53        STR 3 .. UPPER BYTE → Y(k-1)
0118  5A        STR A .. UPPER BYTE → OPERAND 1
0119  ;         ..
0119  ;         ..YAW AXIS COMPUTATIONS FOLLOW
0119  ;         ..
0119  ;         ..FILTERED YAW RATE X GAIN FOLLOWS
0119  F8BA      LDI BA .. BA → D
011B  A3        PLO 3 .. BA → R3.0, (POINTER TO OPERAND 1)
011C  F8BB      LDI BB .. BB → D
011E  A4        PLO 4 .. BB → R4.0, (POINTER TO OPERAND 2)
011F  F875      LDI 75 .. 75 → D
0121  54        STR 4 .. 75 → M(R4), (GAIN VALUE 75 → OPERAND 2)
0122  D7        SEP 7 .. GO TO MULTIPLY SUBROUTINE
0123  ;         ..
0123  ;         ..YAW PENDULUM X GAIN FOLLOWS
0123  F8AB      LDI AB .. AB → D
0125  AA        PLO A .. AB → RA.0, (POINTER TO YAW PENDULUM INPUT)
0126  13        INC 3 .. POSITION POINTER TO OPERAND 1
0127  13        INC 3 ..  "
0128  4A        LDA A .. RETRIEVE YAW PENDULUM INPUT
0129  53        STR 3 .. YAW PENDULUM INPUT → OPERAND 1
012A  14        INC 4 .. POSITION POINTER TO OPERAND 2
012B  14        INC 4 ..  "
012C  F817      LDI 17 .. 17 → D
012E  54        STR 4 .. 17 → M(R4), (GAIN VALUE 17 → OPERAND 2)
```

```
012F  D7        SEP 7 .. GO TO MULTIPLY SUBROUTINE
0130  ;         ..
0130  ;         .. SUMMATION OF INNER AND OUTER LOOPS FOLLOWS
0130  23        DEC 3 .. POSITION POINTER TO OPERAND 1
0131  D9        SEP 9 .. GO TO ADD SUBROUTINE
0132  ;         ..
0132  ;         .. SAVING OF SUM IN TEMP (A9-AA) FOLLOWS
0132  F8A9      LDI A9 .. A9 → D
0134  A3        PLO 3 .. A9 → R3.0, (POINTER TO TEMP UPPER BYTE)
0135  44        LDA 4 .. RETRIEVE UPPER BYTE
0136  53        STR 3 .. UPPER BYTE → TEMP (A9)
0137  13        INC 3 .. POSITION POINTER TO TEMP LOWER BYTE (AA)
0138  04        LDN 4 .. RETRIEVE LOWER BYTE
0139  53        STR 3 .. LOWER BYTE → TEMP (AA)
013A  ;         ..
013A  ;         .. SUM - MAX SUM (65FF) FOLLOWS
013A  F8BA      LDI BA .. BA → D
013C  A3        PLO 3 .. BA → R3.0 (POINTER TO OPERAND 1 UPPER BYTE)
013D  F865      LDI 65 .. 65 → D
013F  53        STR 3 .. 65 → M(R3), (65 → OPERAND 1 UPPER BYTE)
0140  13        INC 3 .. POSITION TO OPERAND 1 LOWER BYTE
0141  F8FF      LDI FF .. FF → D
0143  53        STR 3 .. FF → M(R3), (FF → OPERAND 1 LOWER BYTE)
0144  D8        SEP 8 .. GO TO SUBTRACT SUBROUTINE
0145  ;         ..
0145  ;         ..CHECK FOR EXCESSIVE SUM FOLLOWS
0145  336F      BDF 6F .. TO MAX IF SUBTRACTION RESULT POS
0147  ;         ..
0147  ;         ..RECALL OF SUM FOLLOWS
0147  F8A9      LDI A9 .. A9 → D
0149  A3        PLO 3 .. A9 → R3.0 (POINTER TO TEMP UPPER BYTE)
014A  43        LDA 3 .. RETRIEVE TEMP UPPER BYTE
014B  54        STR 4 .. UPPER BYTE → OPERAND 1
014C  14        INC 4 .. POSITION POINTER TO OPERAND 2
014D  03        LDN 3 .. RETRIEVE TEMP LOWER BYTE
014E  54        STR 4 .. LOWER BYTE → OPERAND 2
014F  ;         ..
014F  ;         ..SUM - MIN SUM (2600) FOLLOWS
014F  F8BA      LDI BA .. BA → D
0151  A3        PLO 3 .. BA → R3.0 (POINTER TO OPERAND 1)
0152  F826      LDI 26 .. 26 → D
0154  53        STR 3 .. 26 → M(R3) (26 → OPERAND 2 UPPER BYTE)
0155  13        INC 3 .. POINTER TO OPERAND 2 LOWER BYTE
0156  F800      LDI 00 .. 00 → D
0158  53        STR 3 .. 00 → M(R3) (00 → OPERAND 2 LOWER BYTE)
0059  D8        SEP 8 .. GO TO SUBTRACT SUBROUTINE
015A  ;         ..
015A  ;         .. CHECK FOR TOO SMALL A SUM FOLLOWS
015A  3B77      BNF 77 .. TO MIN IF SUBTRACTION NEG
015C  ;         ..
015C  ;         .. DIVISION OF SUM BY 64 (BY BINARY POINT SHIFTING) FOLLOWS
```

```
015C  14        INC 4 .. POSITION POINTER TO LOWER BYTE
015D  04        LDN 4 .. RETRIEVE LOWER BYTE
015E  FE        SHL .. SHIFT LEFT
015F  54        STR 4 .. REPLACE LOWER BYTE
0160  24        DEC 4 .. POSITION POINTER TO UPPER BYTE
0161  04        LDN 4 .. RETRIEVE UPPER BYTE
0162  7E        SHLC .. SHIFT LEFT WITH CARRY-IN
0163  54        STR 4 .. REPLACE UPPER BYTE
0164  14        INC 4 .. POSITION POINTER TO LOWER BYTE
0165  04        LDN 4 .. RETRIEVE LOWER BYTE
0166  FE        SHL .. SHIFT LEFT
0167  24        DEC 4 .. POSITION POINTER TO UPPER BYTE
0168  04        LDN 4 .. RETRIEVE UPPER BYTE
0169  7E        SHLC .. SHIFT LEFT WITH CARRY-IN
016A  ;         ..
016A  FDFF      SDI FF .. INVERT (NEGATIVE FEEDBACK)
016C  ;         ..
016C  54        STR 4 .. PUT RESULT IN LOCATION BC FOR OUTPUT
016D  307D      BR 7D .. TO OUT
016F  ;         ..
016F  ;         .. MAX FOLLOWS
016F  F8BC      LDI BC .. BC → D
0171  A4        PLO 4 .. BC → R4.0 (POINTER TO OUTPUT DATA)
0172  F800      LDI 00 .. 00 → D (00 SINCE INVERSION NEEDED)
0174  54        STR 4 .. 00 → OUTPUT DATA
0175  307D      BR 7D .. TO OUT
0177  ;         ..
0177  ;         .. MIN FOLLOWS
0177  F8BC      LDI BC .. BC → D
0179  A4        PLO 4 .. BC → R4.0 (POINTER TO OUTPUT DATA)
017A  F8FF      LDI FF .. FF → D (FF SINCE INVERSION NEEDED)
017C  54        STR 4 .. FF → OUTPUT DATA
017D  ;         ..
017D  ;         .. OUT FOLLOWS
017D  E4        SEX 4 .. RESET X
017E  63        OUT 3 .. OUTPUT YAW COMMAND
017F  ;         ..
017F  ;         .. ROLL AXIS FOLLOWS
017F  ;         ..
017F  ;         .. ROLL RATE X GAIN FOLLOWS
017F  F8BA      LDI BA .. BA → D
0181  A3        PLO 3 .. BA → R3.0 (POINTER TO OPERAND 1)
0182  4A        LDA A .. RETRIEVE ROLL RATE
0183  53        STR 3 .. ROLL RATE → OPERAND 1
0184  F8BB      LDI BB .. BB → D
0186  A4        PLO 4 .. BB → R4.0 (POINTER TO OPERAND 2)
0187  F81D      LDI 1D .. 1D → D
0189  54        STR 4 .. 1D → M(R4) (GAIN VALUE 1D →OPERAND 2)
018A  D7        SEP 7 .. GO TO MULTIPLY SUBROUTINE
018B  ;         ..
```

```
018B  ;          .. ROLL GYRO X GAIN FOLLOWS
018B  13         INC 3 .. POSITION POINTER TO OPERAND 1
018C  13         INC 3 ..  "
018D  4A         LDA A .. RETRIEVE ROLL GYRO INPUT
018E  53         STR 3 .. ROLL GYRO INPUT → OPERAND 1
018F  14         INC 4 .. POSITION POINTER TO OPERAND 2
0190  14         INC 4 ..  "
0191  F866       LDI 66 .. 66 → D
0193  54         STR 4 .. 66 → M(R4), (GAIN VALUE 66 → OPERAND 2)
0194  D7         SEP 7 .. GO TO MULTIPLY SUBROUTINE
0195  ;          ..
0195  ;          .. SUMMATION OF INNER AND OUTER LOOPS FOLLOWS
0195  23         DEC 3 .. POSITION POINTER TO OPERAND 1
0196  D9         SEP 9 .. GO TO ADD SUBROUTINE
0197  ;          ..
0197  ;          .. SAVING OF SUM IN TEMP (A9-AA) FOLLOWS
0197  F8A9       LDI A9 .. A9 → D
0199  A3         PLO 3 .. A9 → R3.0 (POINTER TO TEMP UPPER BYTE)
019A  44         LDA 4 .. RETRIEVE UPPER BYTE
019B  53         STR 3 .. UPPER BYTE → TEMP (A9)
019C  13         INC 3 .. POSITION POINTER TO TEMP LOWER BYTE (AA)
019D  04         LDN 4 .. RETRIEVE LOWER BYTE
019E  53         STR 3 .. LOWER BYTE → TEMP (AA)
019F  ;          ..
019F  ;          .. SUM - MAX SUM (617F) FOLLOWS
019F  F8BA       LDI BA .. BA → D
01A1  A3         PLO 3 .. BA → R3.0 (POINTER TO OPERAND 1 UPPER BYTE)
01A2  F861       LDI 61 .. 61 → D
01A4  53         STR 3 .. 61 → M(R3) (61 → OPERAND 1 UPPER BYTE)
01A5  13         INC 3 .. POSITION POINTER TO OPERAND 1 LOWER BYTE
01A6  F87F       LDI 7F .. 7F → D
01A8  53         STR 3 .. 7F → M(R3), (7F → OPERAND 1 LOWER BYTE)
01A9  D8         SEP 8 .. GO TO SUBTRACT SUBROUTINE
01AA  ;          ..
01AA  ;          .. CHECK FOR EXCESSIVE SUM FOLLOWS
01AA  33D4       BDF D4 .. TO MAX IF SUBTRACTION RESULT POS
01AC  ;          ..
01AC  ;          .. RECALL OF SUM FOLLOWS
01AC  F8A9       LDI A9 .. A9 → D
01AE  A3         PLO 3 .. A9 → R3.0 (POINTER TO TEMP UPPER BYTE)
01AF  43         LDA 3 .. RETRIEVE TEMP UPPER BYTE
01B0  54         STR 4 .. UPPER BYTE → OPERAND 1
01B1  14         INC 4 .. POSITION POINTER TO OPERAND 2
01B2  03         LDN 3 .. RETRIEVE TEMP LOWER BYTE
01B3  54         STR 4 .. LOWER BYTE → OPERAND 2
01B4  ;          ..
01B4  ;:         .. SUM - MIN SUM (2180) FOLLOWS
01B4  F8BA       LDI BA .. BA → D
01B6  A3         PLO 3 .. BA → R3.0 (POINTER TO OPERAND 1)
01B7  F821       LDI 21 .. 21 → D
```

```
01B9  53        STR 3 .. 21 → M(R3) (21 → OPERAND 2 UPPER BYTE)
01BA  13        INC 3 .. POINTER TO OPERAND 2 LOWER BYTE
01BB  F880      LDI 80 .. 80 → D
01BD  53        STR 3 .. 80 → M(R3), (80 → OPERAND 2 LOWER BYTE)
01BE  D8        SEP 8 .. GO TO SUBTRACT SUBROUTINE
01BF  ;         ..
01BF  ;         .. CHECK FOR TOO SMALL A SUM FOLLOWS
01BF  3BDC      BNF DC .. TO MIN IF SUBTRACTION NEG
01C1  ;         ..
01C1  ;         .. DIVISION OF SUM BY 64 (BY BINARY POINT SHIFTING)
01C1  ;         ..FOLLOWS
01C1  14        INC 4 .. POSITION POINTER TO LOWER BYTE
01C2  04        LDN 4 .. RETRIEVE LOWER BYTE
01C3  FE        SHL .. SHIFT LEFT
01C4  54        STR 4 .. REPLACE LOWER BYTE
01C5  24        DEC 4 .. POSITION POINTER TO UPPER BYTE
01C6  04        LDN 4 .. RETRIEVE UPPER BYTE
01C7  7E        SHLC .. SHIFT LEFT WITH CARRY-IN
01C8  54        STR 4 .. REPLACE UPPER BYTE
01C9  14        INC 4 .. POSITION POINTER TO LOWER BYTE
01CA  04        LDN 4 .. RETRIEVE LOWER BYTE
01CB  FE        SHL .. SHIFT LEFT
01CC  24        DEC 4 .. POSITION POINTER TO UPPER BYTE
01CD  04        LDN 4 .. RETRIEVE UPPER BYTE
01CE  7E        SHLC .. SHIFT LEFT WITH CARRY-IN
01CF  ;         ..
01CF  FDFF      SDI FF .. INVERT (NEGATIVE FEEDBACK)
01D1  ;         ..
01D1  54        STR 4 .. PUT RESULT IN LOCATION BC FOR OUTPUT
01D2  3QE2      BR E2 .. TO OUT
01D4  ;         ..
01D4  ;         .. MAX FOLLOWS
01D4  F8BC      LDI BC .. BC → D
01D6  A4        PLO 4 .. BC → R4.0 (POINTER TO OUTPUT DATA)
01D7  F800      LDI 00 .. 00 → D (00 SINCE INVERSION NEEDED)
01D9  54        STR 4 .. 00 →OUTPUT DATA
01DA  30E2      BR E2 .. TO OUT
01DC  ;         ..
01DC  ;         .. MIN FOLLOWS
01DC  F8BC      LDI BC .. BC → D
01DE  A4        PLO 4 .. BC → R4.0 (POINTER TO OUTPUT DATA)
01DF  F8FF      LDI FF .. FF →  D (FF SINCE INVERSION NEEDED)
01E1  54        STR 4 .. FF → OUTPUT DATA
01E2  ;         ..
01E2  ;         .. OUT FOLLOWS
01E2  E4        SEX 4 .. RESET X
01E3  65        OUT 5 .. OUTPUT ROLL COMMAND
01E4  ;         ..
01E4  ;         .. PITCH AXIS FOLLOWS
01E4  ;         ..
```

```
01E4  ;          .. PITCH RATE X GAIN FOLLOWS
01E4  F8BA       LDI BA .. BA → D
01E6  A3         PLO 3 .. BA → R3.0 (POINTER TO OPERAND 1)
01E7  4A         LDA A .. RETRIEVE PITCH RATE
01E8  53         STR 3 .. PITCH RATE → OPERAND 1
01E9  F8BB       LDI BB .. BB → D
01EB  A4         PLO 4 .. BB → R4.0 (POINTER TO OPERAND 2)
01EC  F81C       LDI 1C .. 1C → D
01EE  54         STR 4 .. 1C → M(R4) (GAIN VALUE 1C → OPERAND 2)
01EF  D7         SEP 7 .. GO TO MULTIPLY SUBROUTINE
01F0  C00400     LBR 0400 .. GO TO NEXT PAGE (PAGE 04)
01F3  00         IDL .. UNUSED
  ↓   "          ".  ..  "
01FD  "          "   ..  "
01FE  00         IDL .. USED BY MONITOR
01FF  00         IDL .. USED BY MONITOR

0400  ;          ..
0400  ;          ..PITCH GYRO X GAIN FOLLOWS
0400  13         INC 3 .. POSITION POINTER TO OPERAND 1
0401  13         INC 3 ..  "
0402  0A         LDN A .. RETRIEVE PITCH GYRO INPUT
0403  53         STR 3 .. PITCH GYRO INPUT → OPERAND 1
0404  14         INC 4 .. POSITION POINTER TO OPERAND 2
0405  14         INC 4 ..  "
0406  F8A3       LDI A3 .. A3 → D
0408  54         STR 4 .. A3 → M(R4) (GAIN VALUE A3 →OPERAND 2)
0409  D7         SEP 7 .. GO TO MULTIPLY SUBROUTINE
040A  ;          ..
040A  ;          .. SUMMATION OF INNER AND OUTER LOOPS FOLLOWS
040A  23         DEC 3 .. POSITION POINTER TO OPERAND 1
040B  D9         SEP 9 .. GO TO ADD SUBROUTINE
040C  ;          ..
040C  ;          .. SAVE OF SUM IN TEMP (A9-AA) FOLLOWS
040C  F8A9       LDI A9 .. A9 → D
040E  A3         PLO 3 .. A9 → R3.0 (POINTER TO TEMP UPPER BYTE)
040F  44         LDA 4 .. RETRIEVE UPPER BYTE
0410  53         STR 3 .. UPPER BYTE → TEMP (A9)
0411  13         INC 3 .. POSITION POINTER TO TEMP LOWER BYTE (AA)
0412  04         LDN 4 .. RETRIEVE LOWER BYTE
0413  53         STR 3 .. LOWER BYTE → TEMP (AA)
0414  ;          ..
0414  ;          .. SUM - MAX SUM (7F7F) FOLLOWS
0414  F8BA       LDI BA .. BA → D
0416  A3         PLO 3 .. BA → R3.0 (POINTER TO OPERAND 1 UPPER BYTE)
0417  F87F       LDI 7F .. 7F → D
0419  53         STR 3 .. 7F → M(R3) (7F → OPERAND 1 UPPER BYTE)
041A  13         INC 3 .. POSITION POINTER TO OPERAND 1 LOWER BYTE
041B  F87F       LDI 7F .. 7F → D
041D  53         STR 3 .. 7F → M(R3) (7F → OPERAND 1 LOWER BYTE)
```

```
041E  D8        SEP 8 .. GO TO SUBTRACT SUBROUTINE
041F  ;         ..
041F  ;         .. CHECK FOR EXCESSIVE SUM FOLLOWS
041F  3349      BDF 49 .. TO MAX IF SUBTRACTION RESULT POS
0421  ;         ..
0421  ;         .. RECALL OF SUM FOLLOWS
0421  F8A9      LDI A9 .. A9 → D
0423  A3        PLO 3 .. A9 → R3.0 (POINTER TO TEMP UPPER BYTE)
0424  43        LDA 3 .. RETRIEVE TEMP UPPER BYTE
0425  54        STR 4 .. UPPER BYTE → OPERAND 1
0426  14        INC 4 .. POSITION POINTER TO OPERAND 2
0427  03        LDN 3 .. RETRIEVE TEMP LOWER BYTE
0428  54        STR 4 .. LOWER BYTE → OPERAND 2
0429  ;         ..
0429  ;         .. SUM - MIN SUM (3F80) FOLLOWS
0429  F8BA      LDI BA .. BA → D
042B  A3        PLO 3 .. BA → R3.0 (POINTER TO OPERAND 1)
042C  F83F      LDI 3F .. 3F → D
042E  53        STR 3 .. 3F → M(R3) (3F → OPERAND 2 UPPER BYTE)
042F  13        INC 3 .. POINTER TO OPERAND 2 LOWER BYTE
0430  F880      LDI 80 .. 80 → D
0432  53        STR 3 .. 80 → M(R3) (80 → OPERAND 2 LOWER BYTE)
0433  D8        SEP 8 .. GO TO SUBTRACT SUBROUTINE
0434  ;         ..
0434  ;         .. CHECK FOR TOO SMALL A SUM FOLLOWS
0434  3B51      BNF 51 .. TO MIN IF SUBTRACTION NEG
0436  ;         ..
0436  ;         .. DIVISION OF SUM BY 64 (BY BINARY POINT SHIFTING)
0436  ;         .. FOLLOWS
0436  14        INC 4 .. POSITION POINTER TO LOWER BYTE
0437  04        LDN 4 .. RETRIEVE LOWER BYTE
0438  FE        SHL .. SHIFT LEFT
0439  54        STR 4 .. REPLACE LOWER BYTE
043A  24        DEC 4 .. POSITION POINTER TO UPPER BYTE
043B  04        LDN 4 .. RETRIEVE UPPER BYTE
043C  7E        SHLC .. SHIFT LEFT WITH CARRY-IN
043D  54        STR 4 .. REPLACE UPPER BYTE
043E  14        INC 4 .. POSITION POINTER TO LOWER BYTE
043F  04        LDN 4 .. RETRIEVE LOWER BYTE
0440  FE        SHL .. SHIFT LEFT
0441  24        DEC 4 .. POSITION POINTER TO UPPER BYTE
0442  04        LDN 4 .. RETRIEVE UPPER BYTE
0443  7E        SHLC .. SHIFT LEFT WITH CARRY-IN
0444  ;         ..
0444  FDFF      SDI FF .. INVERT (NEGATIVE FEEDBACK)
0446  ;         ..
0446  54        STR 4 .. PUT RESULT TO LOCATION BC FOR OUTPUT
0447  3057      BR 57 .. TO OUT
0449  ;         ..
0449  ;         .. MAX FOLLOWS
```

```
0449  F8BC      LDI BC .. BC → D
044B  A4        PLO 4 .. BC → R4.0 (POINTER TO OUTPUT DATA)
044C  F800      LDI 00 .. 00 →  D (00 SINCE INVERSION NEEDED)
044E  54        STR 4 .. 00 →  OUTPUT DATA
044F  3057      BR 57 .. TO OUT
0451  ;         ..
0451  ;         .. MIN FOLLOWS
0451  F8BC      LDI BC .. BC →  D
0453  A4        PLO 4 .. BC → R4.0 (POINTER TO OUTPUT DATA)
0454  F8FF      LDI FF .. FF →  D (FF SINCE INVERSION NEEDED)
0456  54        STR 4 .. FF → OUTPUT DATA
0457  ;         ..
0457  ;         .. OUT FOLLOWS
0457  E4        SEX 4 .. RESET X
0458  66        OUT 6 .. OUTPUT PITCH COMMAND
0459  365D      B3 5D .. HAS "T" TIMER EXPIRED
045B  3059      BR 59 .. IF NOT TEST AGAIN
045D  C0000D    LBR 000E .. BRANCH TO BEGINNING OF LOOP
0460  00        IDL .. UNUSED
 ↓    "         "   ..  "
04A8  "         "   ..  "
04A9  00        IDL .. TEMPORARY STORAGE
04AA  00        IDL .. TEMPORARY STORAGE
04AB  00        IDL .. STORAGE FOR YAW PENDULUM INPUT
04AC  00        IDL .. STORAGE FOR ROLL RATE INPUT
04AD  00        IDL .. STORAGE FOR ROLL GYRO INPUT
04AE  00        IDL .. STORAGE FOR PITCH RATE INPUT
04AF  00        IDL .. STORAGE FOR PITCH GYRO INPUT
04B0  00        IDL .. STORAGE FOR X(k)
04B1  00        IDL .. STORAGE FOR X(k-1)
04B2  00        IDL .. STORAGE FOR X(k-2)
04B3  00        IDL .. STORAGE FOR Y(k-1)
04B4  00        IDL .. STORAGE FOR Y(k-2
04B5  00        IDL .. STORAGE FOR Y(k-1) FRAC
04B6  00        IDL .. STORAGE FOR Y(k-2) FRAC
04B7  00        IDL .. STACK
04B8  00        IDL .. STACK
04B9  00        IDL .. STACK
04BA  00        IDL .. USED FOR ARITHMETIC OPERANDS
04BB  00        IDL .. USED FOR ARITHMETIC OPERANDS
04BC  00        IDL .. USED FOR ARITHMETIC OPERANDS
04BD  00        IDL .. USED FOR ARITHMETIC OPERANDS
04BE  ;         ..
04BE  ;         .. INTERRUPT ROUTINE FOR DATA INPUT FOLLOWS.(EXECUTION
04BE  ;         .. STARTS AT 04C0)
04BE  42        LDA 2 .. RESTORE ACCUMULATOR
04BF  70        RET .. RESTORE X AND P AND RETURN TO MAIN PROGRAM
04C0  ;         ..
04C0  22        DEC 2 .. DECREMENT STACK POINTER
04C1  78        SAV .. SAVE STATUS OF X AND P
```

```
04C2  22      DEC 2 .. DECREMENT STACK POINTER
04C3  52      STR 2 .. SAVE ACCUMULATOR
04C4  EA      SEX A .. RESET X
04C5  6B      INP 3 .. INPUT DATA
04C6  E2      SEX 2 .. RESET X
04C7  30BE    BR BE .. BRANCH SO PROGRAM COUNTER LEFT AT BEGINNING
04C9  ;       ..
04C9  ;       .. ADD SUBROUTINE FOLLOWS (EXECUTION STARTS AT 04CA)
04C9  D0      SEP 0 .. RETURN TO MAIN PROGRAM
04CA  ;       ..
04CA  E4      SEX 4 .. RESET X
04CB  03      LDN 3 .. RETRIEVE OPERAND 1 LOWER BYTE
04CC  F4      ADD .. ADD LOWER BYTES
04CD  54      STR 4 .. REPLACE LOWER BYTE SUM
04CE  24      DEC 4 .. POSITION POINTER TO OPERAND 2 UPPER BYTE
04CF  23      DEC 3 .. POSITION POINTER TO OPERAND 1 UPPER BYTE
04D0  03      LDN 3 .. RETRIEVE OPERAND 1 UPPER BYTE
04D1  74      ADC .. ADD UPPER BYTES PLUS CARRY
04D2  54      STR 4 .. REPLACE UPPER BYTE SUM
04D3  30C9    BR C9 .. BRANCH SO PROGRAM COUNTER LEFT AT BEGINNING
04D5  ;       ..
04D5  ;       .. SUBTRACT SUBROUTINE FOLLOWS (EXECUTION STARTS AT 04D6)
04D5  D0      SEP 0 .. RETURN TO MAIN PROGRAM
04D6  E4      SEX 4 .. RESET X
04D7  03      LDN 3 .. RETRIEVE OPERAND 1 LOWER BYTE
04D8  F5      SD .. SUBTRACT LOWER BYTES
04D9  54      STR 4 .. REPLACE LOWER BYTE RESULT
04DA  24      DEC 4 .. POSITION POINTER TO OPERAND 2 UPPER BYTE
04DB  23      DEC 3 .. POSITION POINTER TO OPERAND 1 UPPER BYTE
04DC  03      LDN 3 .. RETRIEVE OPERAND 1 UPPER BYTE
04DD  75      SDBI .. SUBTRACT UPPER BYTES INCLUDING BORROW
04DE  54      STR 4 .. REPLACE UPPER BYTE RESULT
04DF  30D5    BR D5 .. BRANCH SO PROGRAM COUNTER LEFT AT BEGINNING
04E1  ;       ..
04E1  ;       .. MULTIPLY SUBROUTINE FOLLOWS (EXECUTION STARTS AT 04E2)
04E1  D0      SEP 0 .. RETURN TO MAIN PROGRAM
04E2  ;       ..
04E2  E3      SEX 3 .. RESET X
04E3  F808    LDI 08 .. 08 → D
04E5  A6      PLO 6 .. 08 → R6.0, INITIALIZE LOOP COUNTER
04E6  F800    LDI 00 .. 00 → D
04E8  B5      PHI 5 .. 00 → R5.1, INITIALIZE ACCUM PROD
04E9  04      LDN 4 .. RETRIEVE MULTIPLIER
04EA  F6      SHR .. SHIFT MULTIPLIER RIGHT
04EB  3BF0    BNF F0 .. BRANCH IF LSB WAS NOT A ONE
04ED  95      GHI 5 .. RETRIEVE ACCUM PROD
04EE  F4      ADD .. ADD MULTIPLICAND AND ACCUM PROD
04EF  38      SKP .. SKIP NEXT INSTRUCTION
04F0  95      GHI 5 .. RETRIEVE ACCUM PROD
04F1  76      SHRC .. SHIFT ACCUM PROD RIGHT
```

```
04F2  B5       PHI 5 .. REPLACE SHIFTED ACCUM PROD
04F3  04       LDN 4 .. RETRIEVE MULTIPLIER
04F4  76       SHRC .. SHIFT MULTIPLIER RIGHT (WITH CARRY)
04F5  54       STR 4 .. REPLACE SHIFTED MULTIPLIER (BECOMES LOWER
04F5  ;        .. BYTE RESULT THE LAST TIME THRU)
04F6  26       DEC 6 .. DECREMENT LOOP COUNTER
04F7  86       GLO 6 .. RETRIEVE LOOP COUNTER
04F8  3AE9     BNZ E9 .. THRU LOOP AGAIN IF LOOP COUNTER NOT ZERO
04FA  95       GHI 5 .. RETRIEVE ACCUM PROD
04FB  53       STR 3 .. ACCUM PROD BECOMES UPPER BYTE RESULT
04FC  30E1     BR E1 .. BRANCH SO PROGRAM COUNTER LEFT AT BEGINNING
04FE  00       IDL .. USED BY MONITOR
04FF  00       IDL .. USED BY MONITOR
```

Vita

PII Redacted

Olin D. Koger was born [redacted], [redacted] He graduated from high school in [redacted] 1961. He then attended Bradley University, where he received a Bachelor of Science Degree in Industrial Engineering in Jan 1966. He concurrently received a commission in the United States Air Force through the ROTC program. He was then assigned as a Minuteman Missile Launch Officer at Malmstrom Air Force Base, Montana. This was followed by an assignment as a spacecraft ground controller in the Defense Meteorlogical Satellite Program at Offutt Air Force Base, Nebraska. A one-year assignment as a BMEWS Space Surveillance Officer at Clear Air Force Station, Alaska preceded his assignment to AFIT in June, 1976, where he completed the Engineering Science and Graduate Electrical Engineering programs.

Permanent address: [redacted]

This thesis was typed by [redacted]

PII Redacted

Vita

Dale E. Tietz was born [redacted]. He graduated from [redacted] in 1967 and thereafter proceeded to the USAF Academy, Colorado. In 1971, he received the degree of Bachelor of Science in Chemistry and was commissioned in the U.S. Air Force. After completion of pilot training at Laughlin AFB, Texas in 1973, he then entered C-141 flight training at McGuire AFB, New Jersey. Four years and 2000 hours later, he received notification to attend the Air Force Institute of Technology, Wright-Patterson AFB, Ohio. Graduation in December 1978 will be the culmination of two more years of study and a Master's Degree in Electrical Engineering.

PII Redacted

Permanent address: [redacted]

PII Redacted

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: AFIT/GE/EE/78D-31
2. GOVT ACCESSION NO.:
3. RECIPIENT'S CATALOG NUMBER:

4. TITLE *(and Subtitle)*: MICROPROCESSOR-BASED DIGITAL AUTOPILOT DEVELOPMENT FOR THE XBQM-106 MINI-RPV

5. TYPE OF REPORT & PERIOD COVERED: MS Thesis

6. PERFORMING ORG. REPORT NUMBER:

7. AUTHOR(s):

Olin D. Koger, Major
Dale E. Tietz, Captain

8. CONTRACT OR GRANT NUMBER(s):

9. PERFORMING ORGANIZATION NAME AND ADDRESS:

Air Force Institute of Technology (AFIT-EN)
Wright-Patterson AFB, Ohio 45433

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS:

11. CONTROLLING OFFICE NAME AND ADDRESS:

Mini-RPV Group (AFFDL/FXG)
Air Force Flight Dynamics Laboratory
Wright-Patterson AFB, Ohio 45433

12. REPORT DATE: December, 1978

13. NUMBER OF PAGES: 252

14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)*:

15. SECURITY CLASS. *(of this report)*: Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE:

16. DISTRIBUTION STATEMENT *(of this Report)*:

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*:

18. SUPPLEMENTARY NOTES:

Approved for public release; IAW AFR 190-17

Joseph P. Hipps, Major, USAF
Director of Information 1-23-79

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*:

Digital Control
Microprocessor
Autopilot
Mini-RPV

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*:

The development of a digital autopilot for the XBQM-106 Mini-Remotely Piloted Vehicle (RPV) is described. The design is based on the RCA CDP-1802 microprocessor with ultra-low component cost as a primary goal. The digital autopilot is developed through the breadboard stage, and performance testing is accomplished via hybrid simulation. First, aircraft transfer functions are derived from aerodynamic wind tunnel data. Then a complete control analysis is made of the existing analog autopilot, sensors, actuators, and aircraft

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

dynamics. This is followed by synthesis of the digital algorithms (including a digital filter) by means of digitization. Hardware is selected based on system requirements, and hardware design details are included. Software to execute the digital algorithms is developed and described. The results of testing by hybrid simulation show that the performance of the digital autopilot is comparable to the existing analog autopilot. Finally, recommendations for the follow-on study are included.